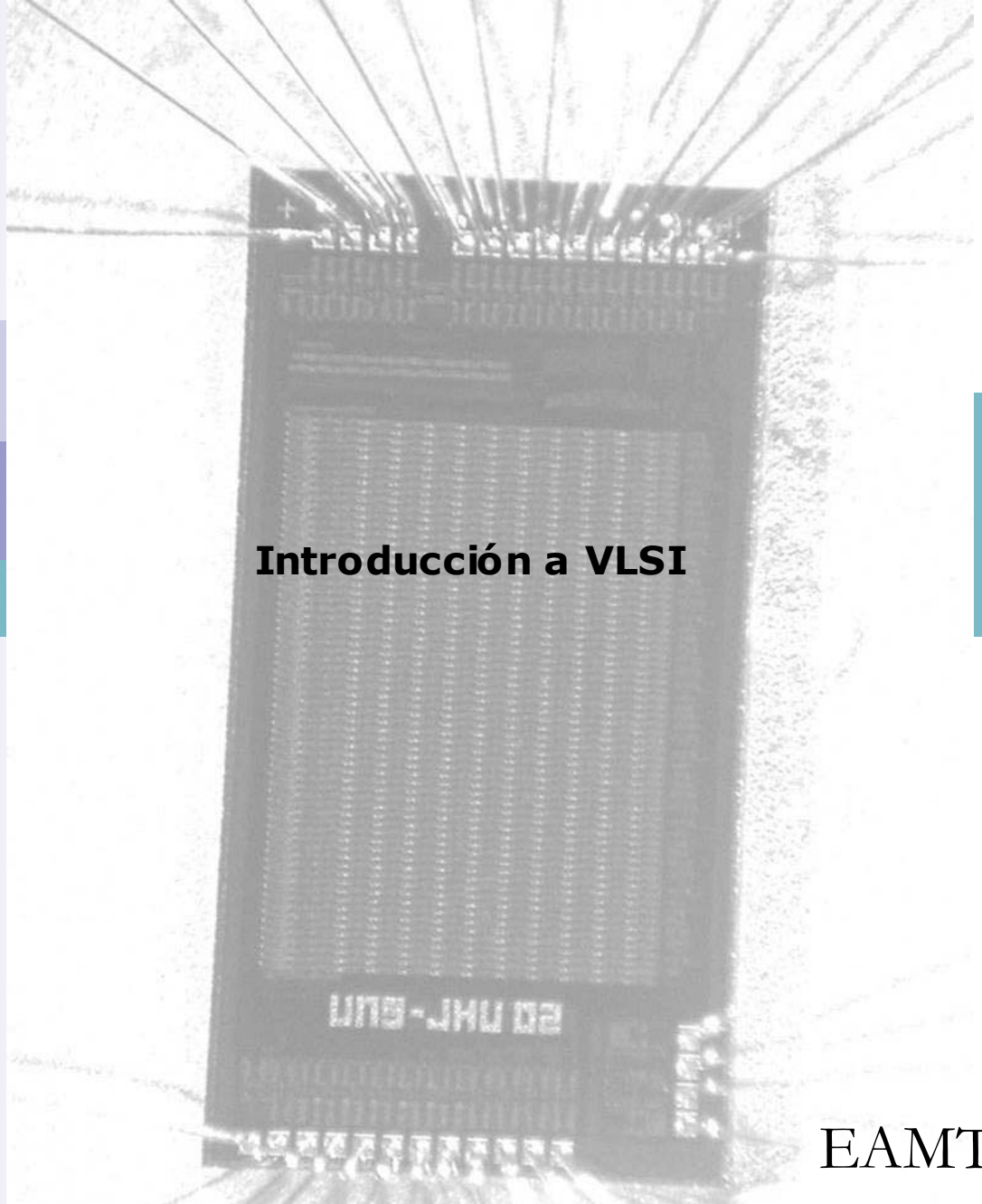
# Introducción a VLSI

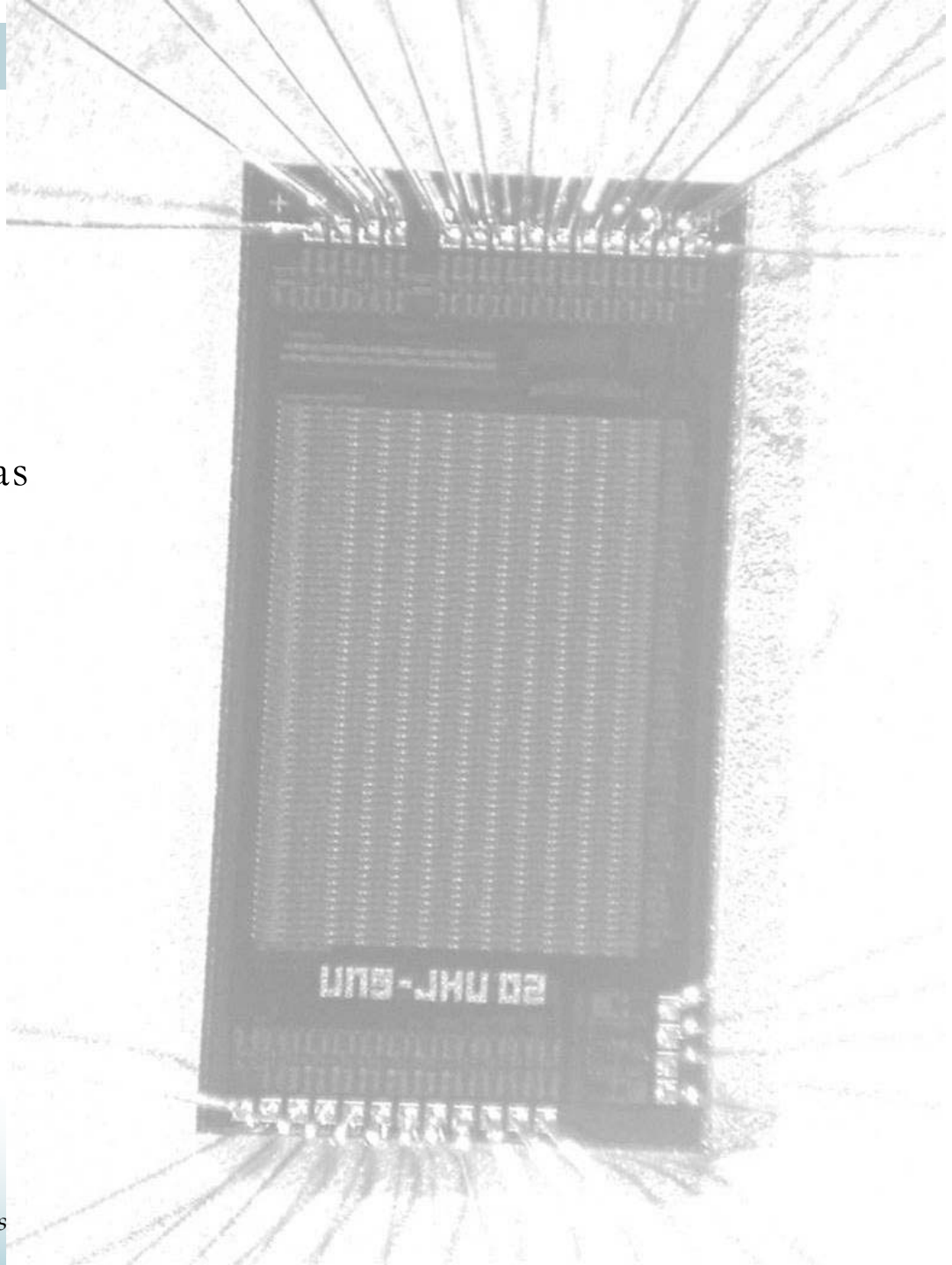


EAMTA 2006


# Clase 3: Lógica Combinacional y Secuencial

# Programa

- El Transistor MOS
- Layers y Layout
- **Lógica Combinacional**
- Lógica Secuencial y Subsistemas

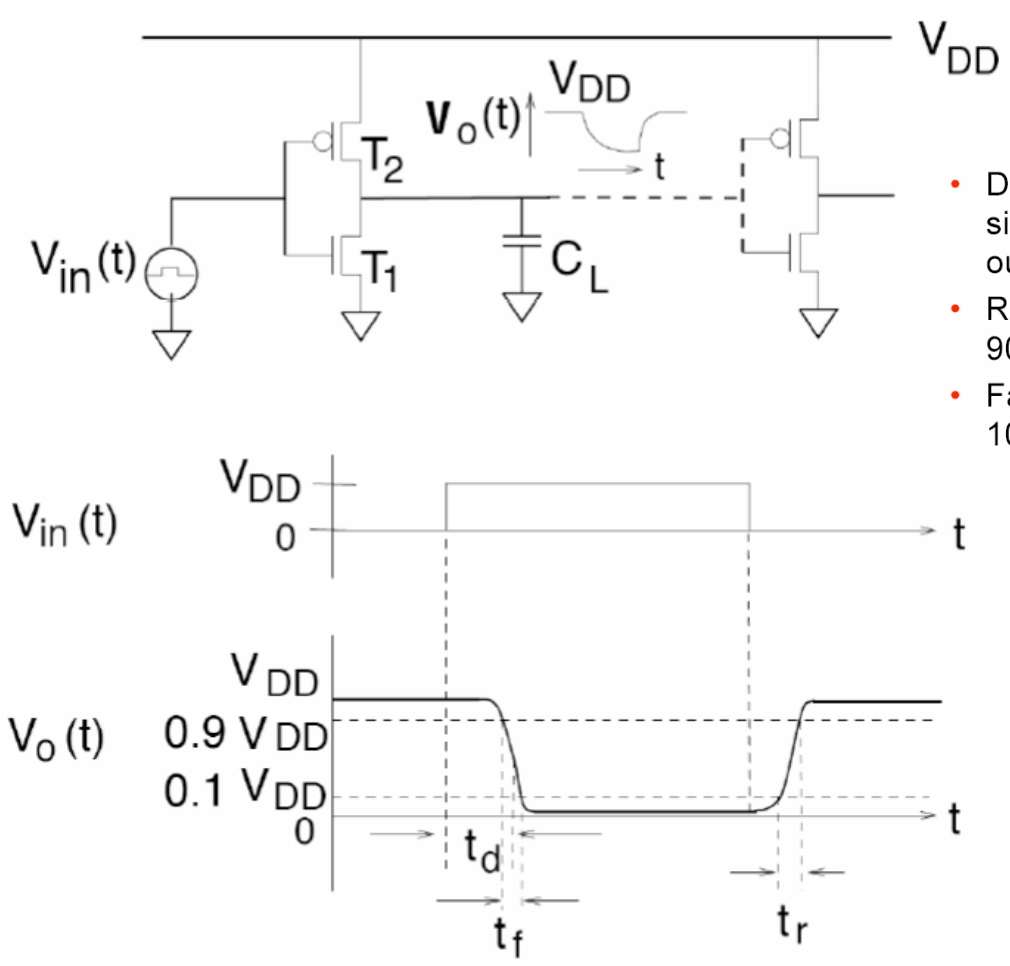


## Organización

- Tiempos de propagación y retardo
- Lógica Combinacional
- Lógica “ratioed”
- Lógica dinámica
- Lógica estática
- Propiedades de compuertas CMOS
- Lógica secuencial

# Propagation delay

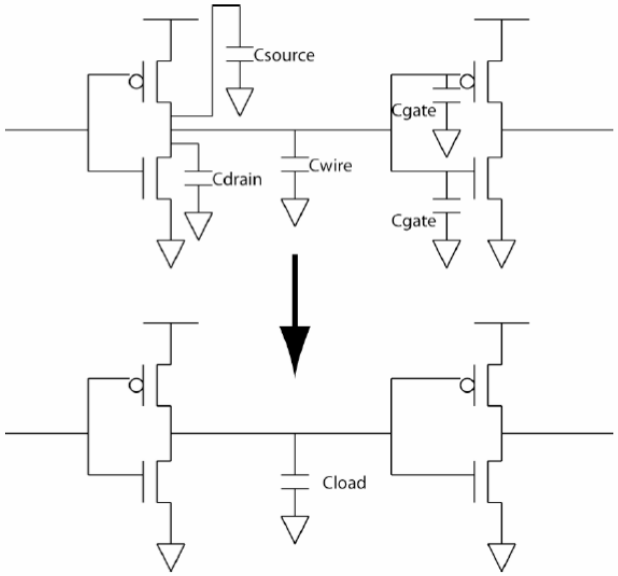
# Delay definitions



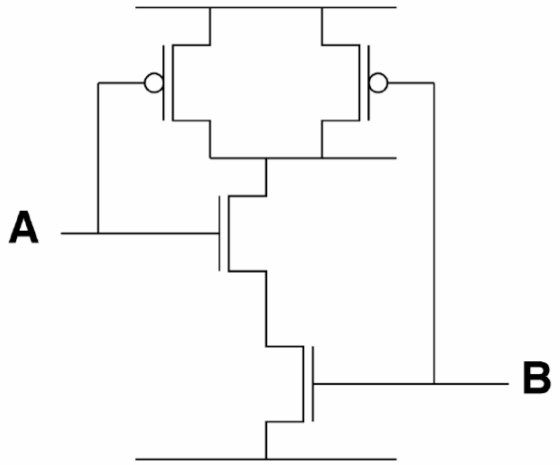

- Delay time ($t_d$ or $t_{pd}$) -- Delay between when the input signal to a gate reaches the 50% point and when the output signal reaches the 50% point.
- Rise time ($t_r$) -- time it takes for a signal to go from 10% to 90% of its output range
- Fall time ($t_f$) -- time it takes for a signal to go from 90% to 10% of its output range

# Delay cause

- Gate delays are determined by how quickly the driving gate can charge/discharge its load capacitance

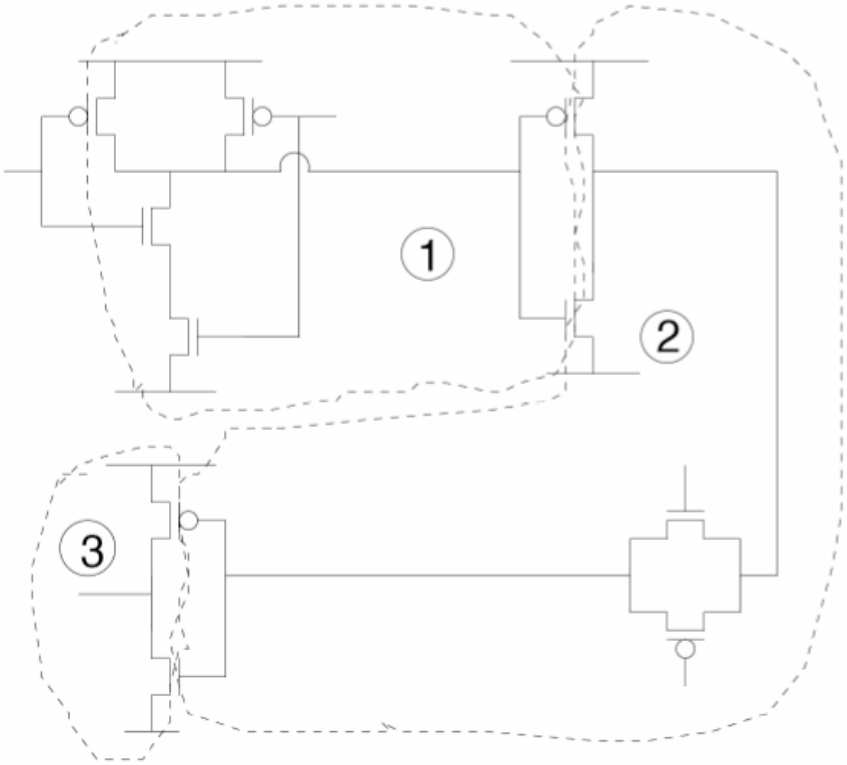



- Gate delay may vary depending on which inputs are changing -- generally use the worst case



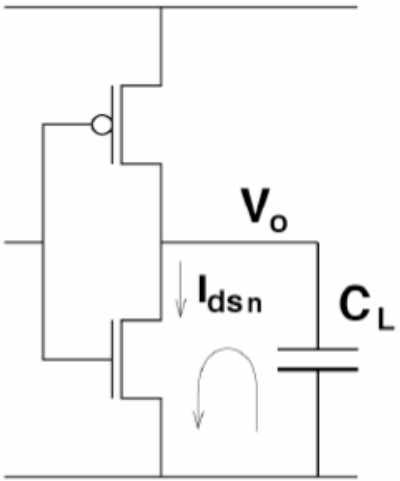
# How to handle a big circuit

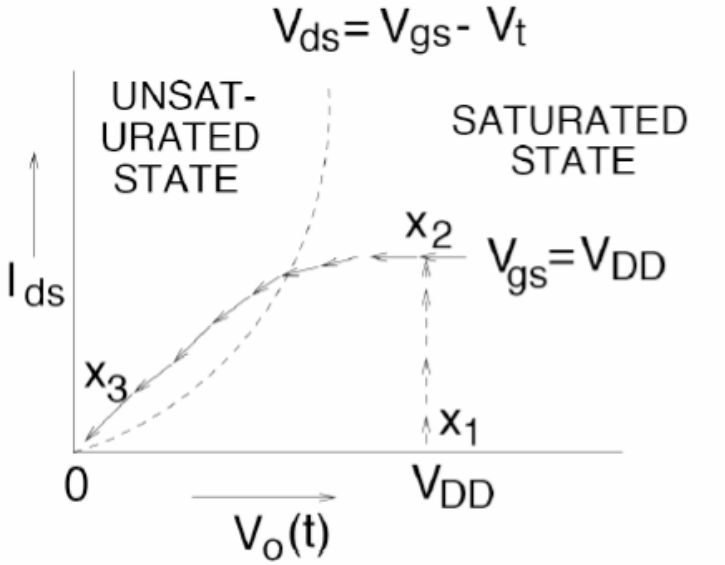
- Divide circuit into *DC-connected components*, solve for each component



# Fall time analysis

During the fall time one or more nMOS transistors discharge the energy stored in the output capacitance



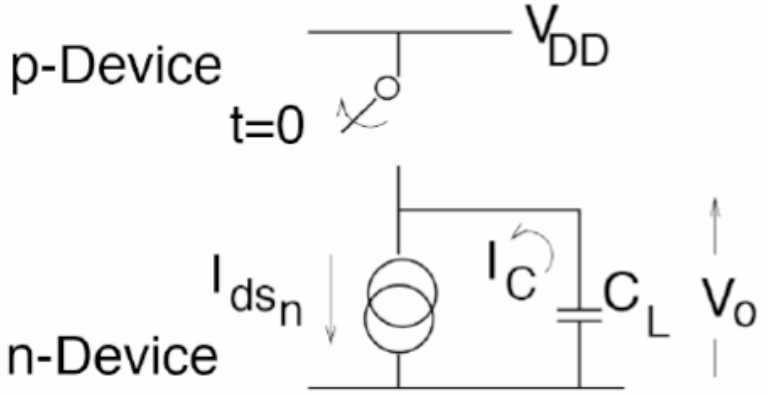




$x_1$ : Initial Operating Point

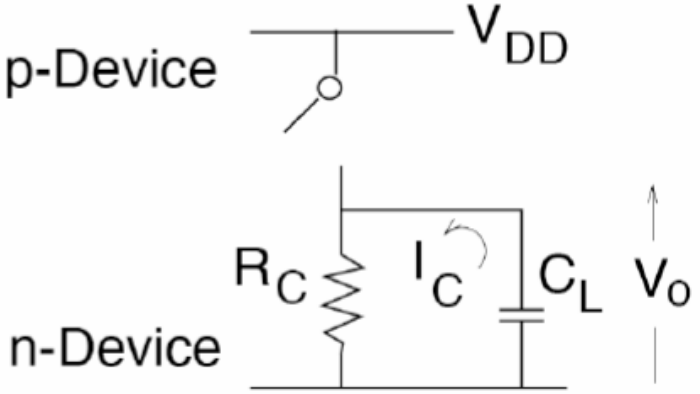
$x_3$ : Operating Point after Completion of Switching

# Fall time analysis

During the fall time, the nMOS transistor starts in the saturated region and passes into the linear region




SATURATION: $V_0 \geq V_{DD} - V_{tn}$

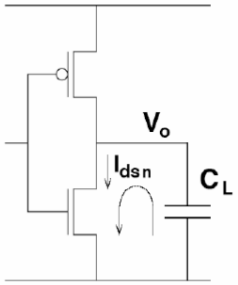



LINEAR: $0 < V_o \leq V_{DD} - V_{tn}$

# Fall time analysis

Divide fall time into two components: $t_{f,sat}$ and $t_{f,linear}$

In saturation, current through the transistor is constant




$$C_L \frac{dV_o}{dt} = -I_{ds_n} = -\frac{\beta_n}{2}(V_{dd} - V_{tn})^2$$

This becomes $$\frac{-2C_L}{\beta_n(V_{dd} - V_{tn})^2} dV_o = dt$$

Define $t_1$, $t_2$ such that $V_o(t_1) = 0.9V_{dd}$ and $V_o(t2) = V_{dd} - V_t$. Then,

$$\frac{-2C_L}{\beta_n(V_{dd} - V_{tn})^2} \int_{0.9V_{dd}}^{V_{dd}-V_{tn}} dV_o = \int_{t_1}^{t_2} dt = t_2 - t_1$$

And $t_{f,\,sat}$ is:

$$\frac{2C_L(V_{tn} - 0.1V_{dd})}{\beta_n(V_{dd} - V_{tn})^2}$$

# Fall time analysis

In the linear region, current through the transistor depends on $V_o$

$$C_L \frac{dV_o}{dt} = -I_{ds_n} = -\beta_n \left[ (V_{dd} - V_{tn}) V_o - \frac{1}{2} V_o^2 \right]$$

And $t_{f,\ linear}$ becomes

$$\frac{C_L}{\beta_n (V_{dd} - V_{tn})} \int_{0.1V_{dd}}^{V_{dd}-V_{tn}} \frac{dV_o}{\frac{V_o^2}{2(V_{dd}-V_{tn})} - V_o}$$

Integrating, we get $t_{f,\ linear} = \frac{C_L}{\beta_n (V_{dd} - V_{tn})} \ln \left( \frac{19V_{dd} - 20V_{tn}}{V_{dd}} \right)$

And

$$t_f = \frac{2C_L}{\beta_n (V_{dd} - V_{tn})} \times \left[ \frac{V_{tn} - 0.1V_{dd}}{V_{dd} - V_{tn}} + \frac{1}{2} \ln \left( \frac{19V_{dd} - 20V_{tn}}{V_{dd}} \right) \right]$$

For many processes, $V_t$ ~= $0.2V_{dd}$, allowing us to approximate

$$t_f \approx \frac{4C_L}{\beta_n V_{dd}}$$

# Rise time analysis

Redoing the same analysis for the pMOS transistor in pull-up gives

$$t_r \approx \frac{4C_L}{\beta_p V_{dd}}$$

Note that beta for pMOS tends to be about 1/2 beta for nMOS given equivalent size devices, so typically want pMOS about twice as wide as nMOS to get equivalent rise and fall times

# Gate delay estimation

Somewhat more complicated -- depends on rise and fall times of the input signals.

Assuming (unrealistically) that the input rises or falls in zero time, then the gate delay can be approximated as half of the rise or fall time for the gate, and averaged to

$$t_d \approx \frac{t_r + t_f}{4}$$

Since gates are generally driven by other gates, we need a better approximation for real circuits

Could just simulate the design, but that wouldn't give much insight

- Simulation is good for verifying that something works the way you want but not for designing it to do so

# Circuit delay estimation

1. Divide circuit into DC-connected blocks as we talked about at the start of lecture.
2. Compute a simple delay model for each block
3. Add the delays for each block to get overall delay.

In CMOS, a DC-connected block (stage) will be either:

1. A single logic gate
2. A transmission-gate network and the gates driving it

In the simple case, a block switches in response to a single signal changing at one of its inputs, called the *trigger* signal

# Simplifying the problem

As we've seen, solving even simple transistor networks analytically requires solving differential equations that change at region boundaries

- For example, the rise and fall time derivations we presented really only apply for inverters, and computing them for larger gates is more complex
- Now think about doing that for a 100-transistor network, much less anything bigger

Solution: *linearize* the transistor equations by replacing each transistor with an equivalent resistor, $R_{eff}$

# Computing Reff

$R_{eff}$ for a transistor depends on what region it's operating in

- Example: $R_{eff} = dV_{ds}/dI_{ds}$ = infinity in saturation
- In linear region, $$\frac{1}{R_{eff}} \triangleq \left.\frac{dI_{ds}}{dV_{ds}}\right|_{V_{ds}=0} \approx \beta(V_{gs} - V_t)$$

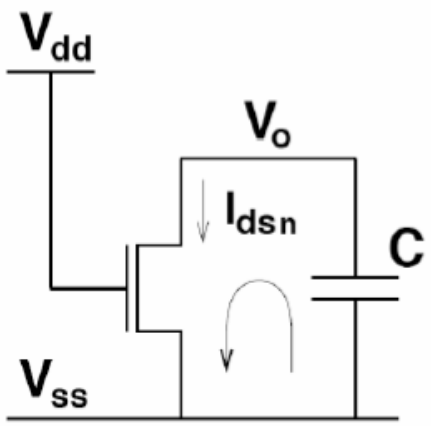
During switching, a transistor travels through different regions of operation

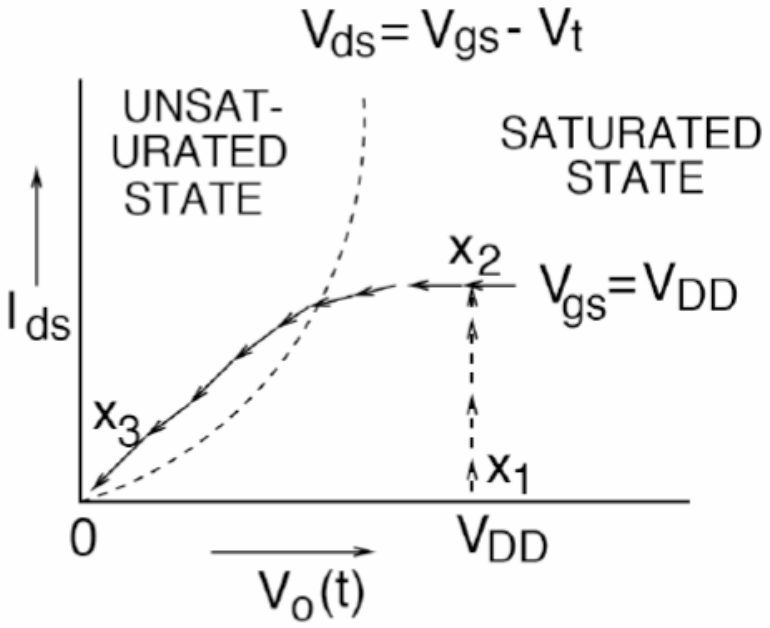
We want a single $R_{eff}$ that approximates the behavior of a transistor during the entire rise/fall time

- $R_{eff}$ will be different depending on whether the transistor is transmitting a zero or a one.

# NMOS transistor, logic 0

nMOS transistor, discharging capacitor to ground



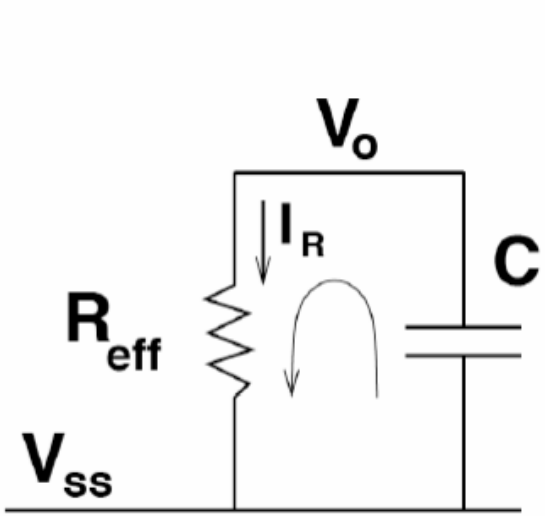



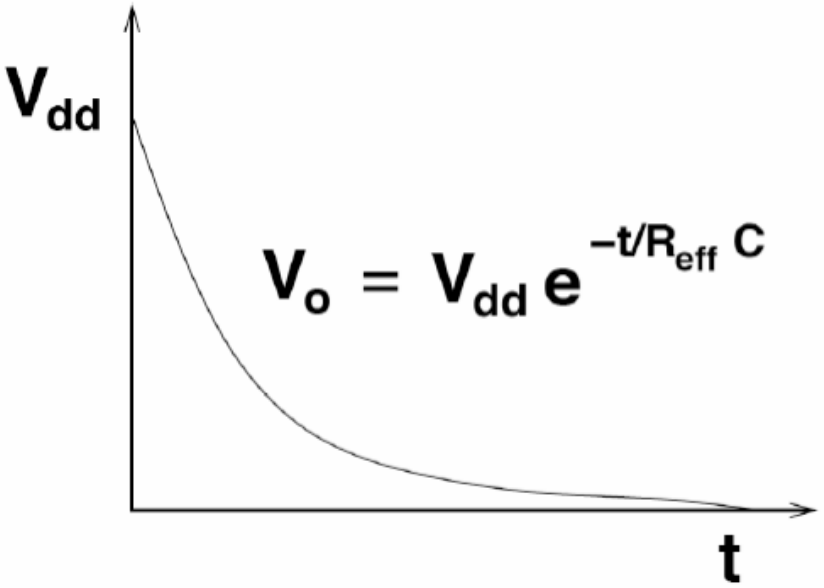

$x_1$ : Initial Operating Point

$x_3$ : Operating Point after Completion of Switching

# NMOS transistor, logic 0

- Want to model as a single network with $R_{eff}$, C



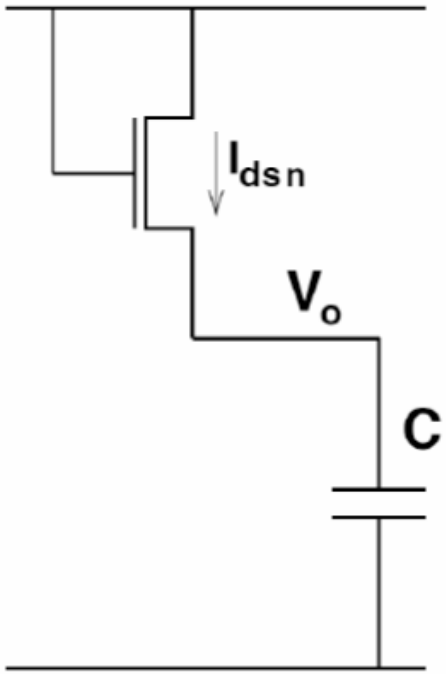



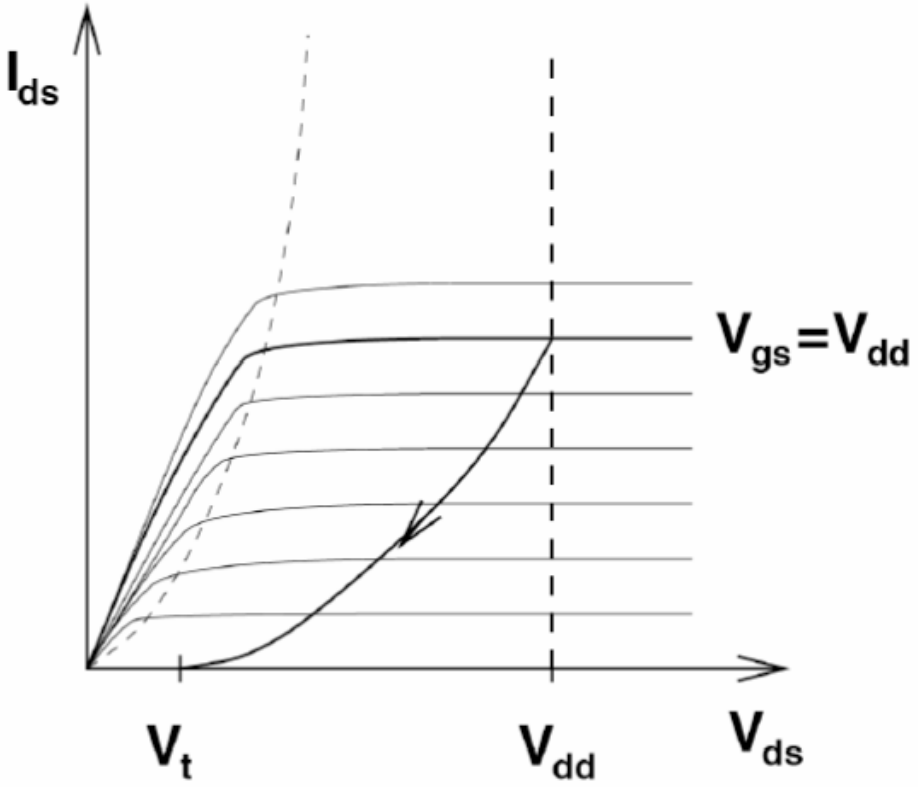

- Choose $R_{eff}$ so that both circuits have the same fall time

$$\frac{4}{\beta_n V_{dd}} C \approx t_f = \ln(9) R_{eff} C \Rightarrow R_{eff} = \frac{4/\ln(9)}{\beta_n V_{dd}} \approx \frac{1.82}{\beta_n V_{dd}}$$

# NMOS transistor, logic 1

Back to the I-V curve



# NMOS transistor, logic 1

Problem: $V_O$ never makes it to 0.9 $V_{dd}$ in this case, assuming $V_t = 0.2V_{dd}$, so can't try to match the full rise time

If we try to match the time it takes to hit $0.5V_{dd}$, we get

$$R_{eff} \approx \frac{25/[6\ln(2)]}{\beta_n V_{dd}} \approx \frac{6.0}{\beta_n V_{dd}}$$

Note that this is about 3x larger than the discharging case. Why does this make sense?

# PMOS transistors

Similar expressions can be derived for a pMOS transistor, using $\beta_p$ instead of $\beta_n$.

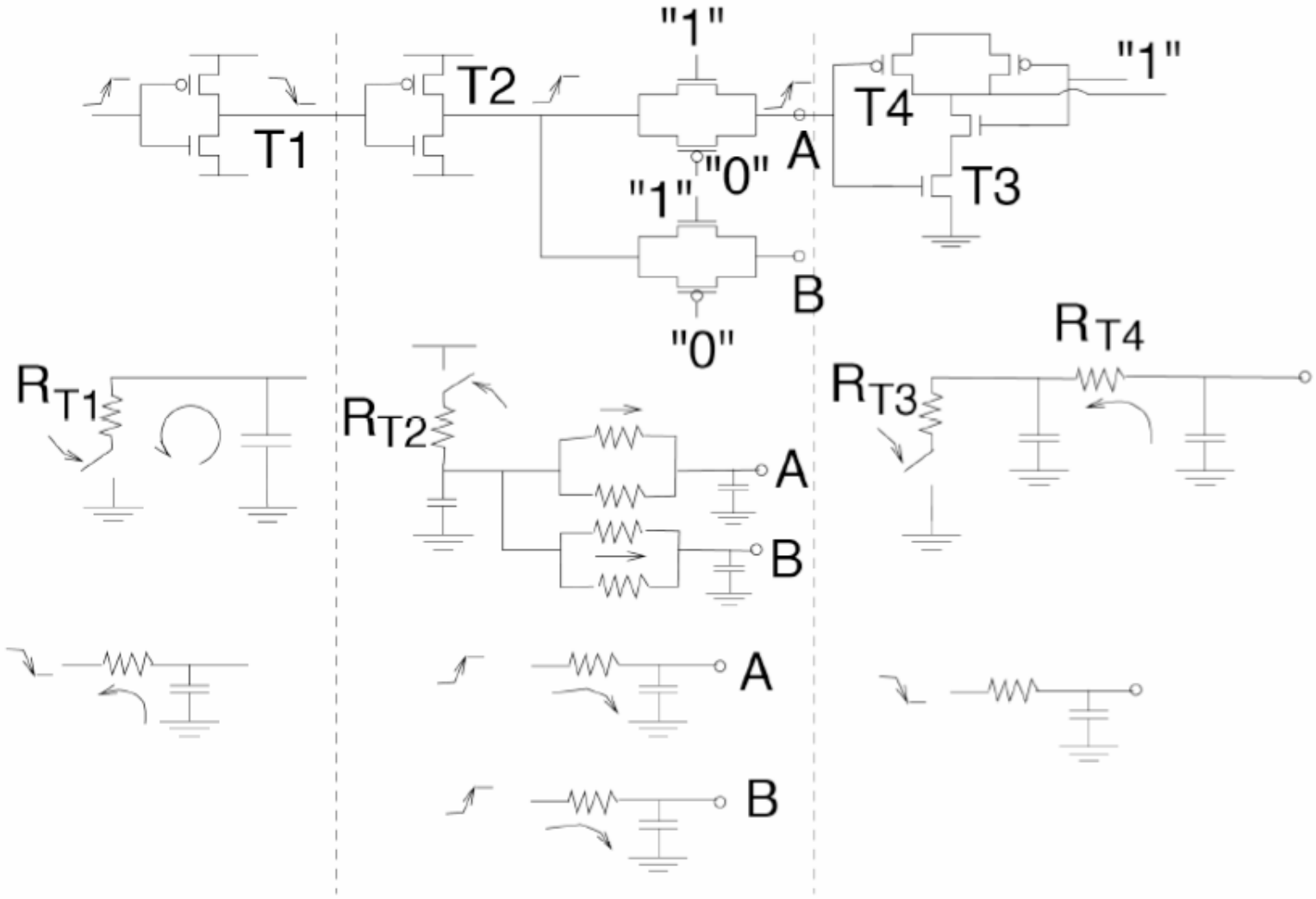
pMOS transmitting a logic 1:

$$R_{eff} = \frac{4/\ln(9)}{\beta_p V_{dd}} \approx \frac{1.82}{\beta_p V_{dd}}$$

pMOS transmitting a logic 0:

$$R_{eff} \approx \frac{25/[6\ln(2)]}{\beta_p V_{dd}} \approx \frac{6.0}{\beta_p V_{dd}}$$

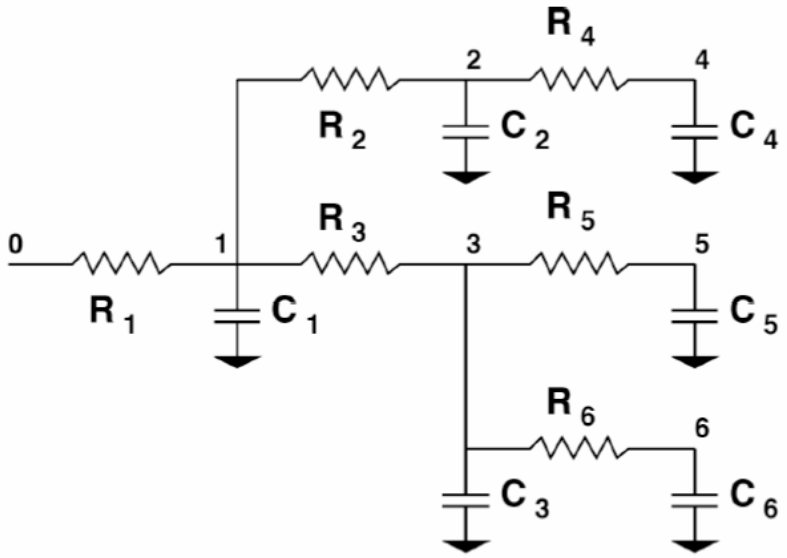
# Example of the process



# Estimating stage delay

Need to decide whether each transistor is off, transmitting a logic 0, or transmitting a logic 1.

Then, can replace the stage with an RC network




This is *still* too complex -- solving these networks exactly requires solving several simultaneous differential equations.

Simplify further by reducing to an equivalent single-RC network with time constant $\tau_{eq} = R_{eq}C_{eq}$

using Elmore Delay formula

$$\tau_1 = R_1 C_1$$
$$\tau_2 = R_1 C_2$$
$$\tau_3 = (R_1 + R_3) C_3$$
$$\tau_4 = R_1 C_4$$
$$\tau_5 = (R_1 + R_3 + R_5) C_5$$
$$\tau_6 = (R_1 + R_3) C_6$$

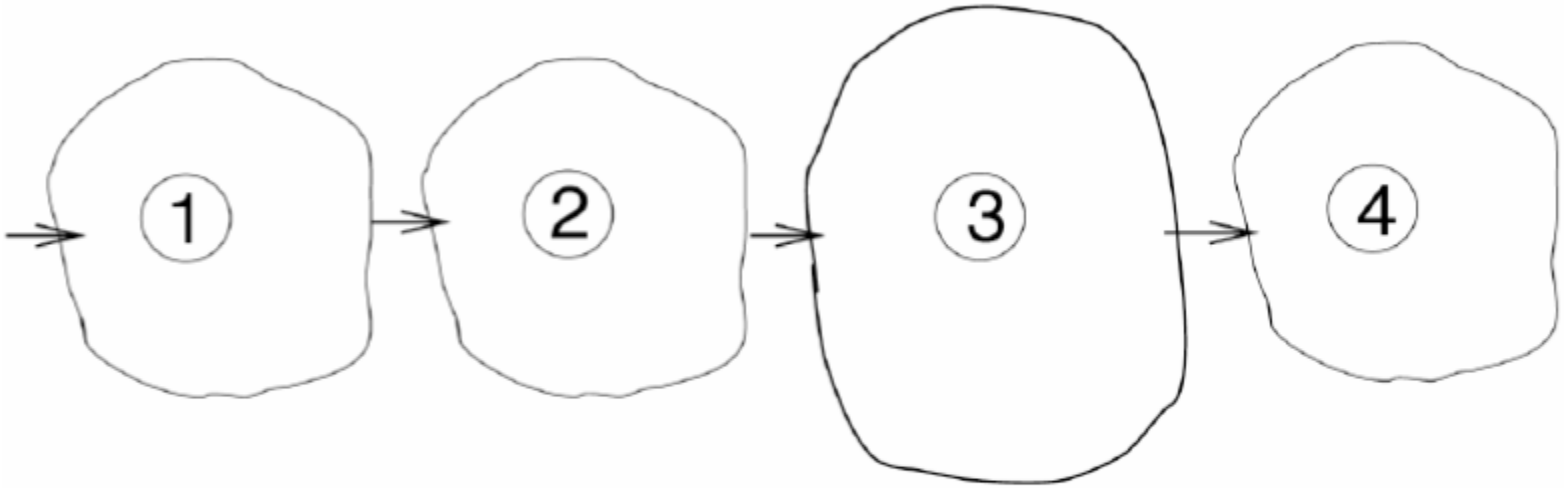
$$\tau_{eq} = \tau_1 + \tau_2 + \tau_3 + \tau_4 + \tau_5 + \tau_6$$

# Elmore delay method

1. Identify an input node and an output node, and the unique path between them. Call the path P
2. For every node n in the tree other than the input node
   1. Identify the unique path between the input node and n
   2. Identify the sub-path that is the intersection of this path and P
   3. Sum the resistances along the sub-path, call it $R_n$
   4. If Cn is the capacitance at n, create a $\tau_n = R_n C_n$
3. Sum the $\tau_{n \text{ to get }} \tau_{eq}$ for the network
4. Given $\tau_{eq}$, can select $R_{eq}$ and $C_{eq}$ values to match, but generally don't need to, $\tau_{eq}$ is enough

# Cascades

Suppose stage 1 feeds stage 2




If the time constant of stage 1 is $\tau_1$, when is stage 2 *triggered*? (I.e., when does it start switching)

Time for yet another approximation

# Cascades

By convention, assume that stage 2 is triggered when the output of stage 1 has completed 90% of its logic swing and is thus well within $V_t$ of one of the rails

Thus, the delay through stage 1 is

$$\tau_{d1} = \ln(10)\tau_{eq1} \approx 2.3\tau_{eq1}$$

Thus, if stage 1 triggers at time 0, stage 2 triggers at time 2.3 $\tau_1$, stage 3 triggers at time 2.3 $\tau_1$ + 2.3 $\tau_2$, etc.

# Device sizing: the inverter

# Device Sizing

- Assuming a symmetrical inverter, the capacitance is composed of:

$$C_L = C_{\text{int}} + C_{ext}$$

  - Cint is the self-load, associated with diffusion and gate-drain (Miller)
  - Cext is extrinsec, load, wiring, etc.

$$t_p = 0.69\, R_{eq}\,(C_{\text{int}} + C_{ext}) = t_{p0}\,(1 + C_{ext} / C_{\text{int}})$$

  - where $t_{p0} = 0.69\, R_{eq} C_{\text{int}}$ is the intrinsec delay (Cext=0)

- What are the consequences of scaling ?

# Device Sizing

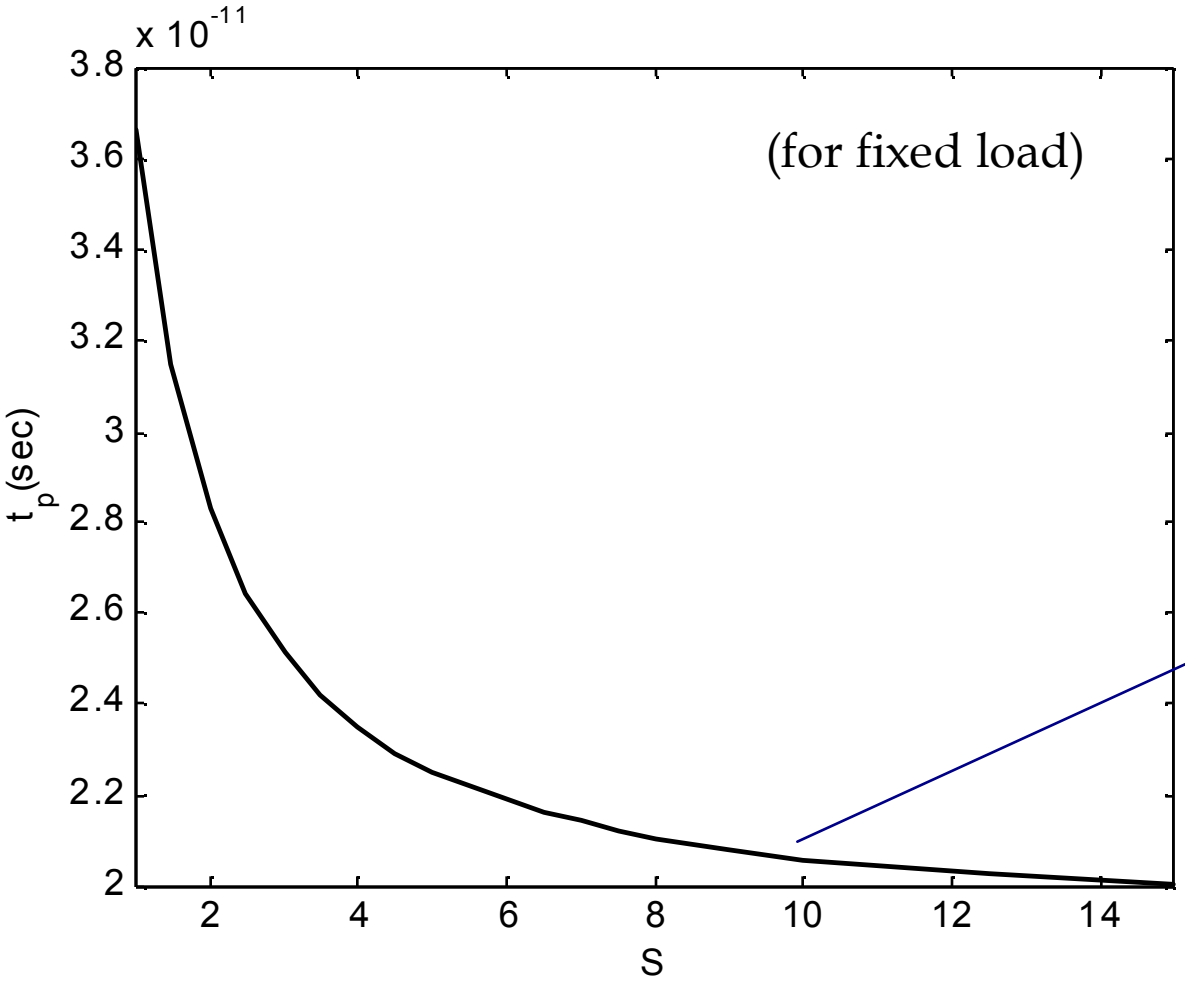
- Cint scales with size ratio S, and also Req:

$$C_{\text{int}} = SC_{iref}, \quad R_{eq} = R_{iref} / S,$$

- The delay is: $t_p = t_{p0}(1 + C_{ext} / SC_{iref})$

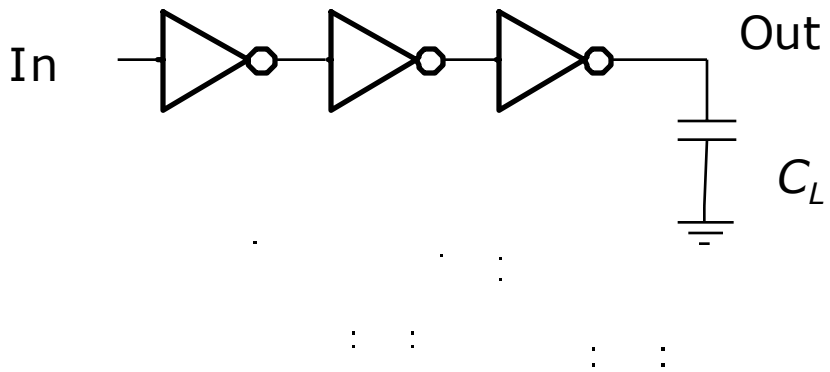
- **Intrinsic delay** is **independent of sizing** and is determined by technology and layout.
- *S* infinitely large gives eliminates the impact of an external load
  - (Yet, a big enough sizing produces similar results with less Silicon area)
- A big inverter has **big input capacitance** and affects the previous stages !

# Device Sizing




Self-loading effect:
Intrinsic capacitances dominate

# Inverter chain sizing




$$Delay \sim R_W\left(C_{int} + C_L\right)$$

$$t_p = kR_W C_{\text{int}}\left(1 + C_L / C_{int}\right) = t_{p0}\left(1 + f/\gamma\right)$$

If $C_L$ is given:
- How many stages are needed to minimize the delay?
- How to size the inverters?

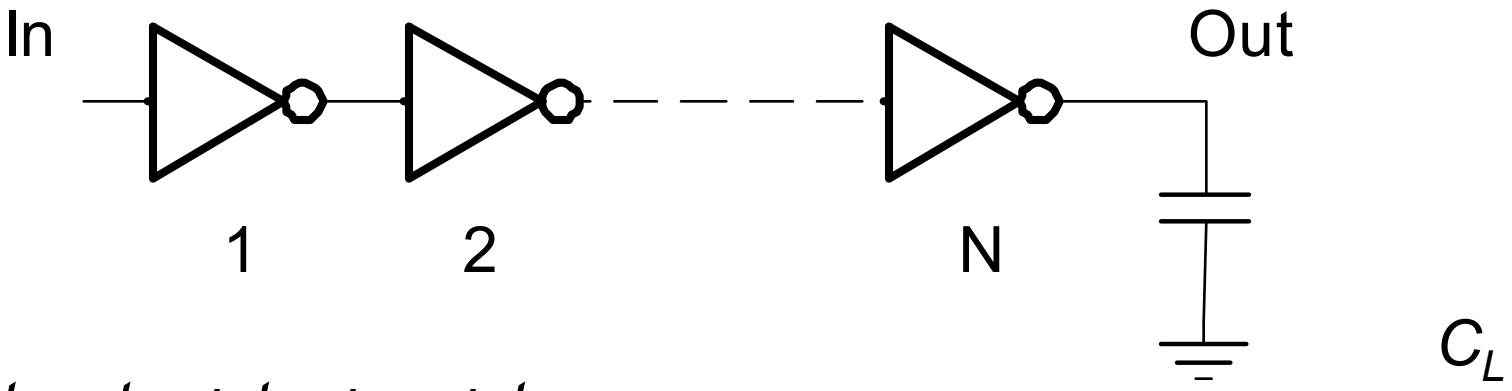
May need some additional constraints.

$C_{int} = \gamma C_{gin}$ with $\gamma \approx 1$

$f = C_L / C_{gin}$ - effective fanout

Delay is only a function of the ratio between its external load capacitance and its input capacitance

# Apply to Inverter Chain




$$t_p = t_{p1} + t_{p2} + \ldots + t_{pN}$$

$$t_{pj} \sim R_{unit} C_{unit} \left( 1 + \frac{C_{gin,j+1}}{\gamma C_{gin,j}} \right)$$

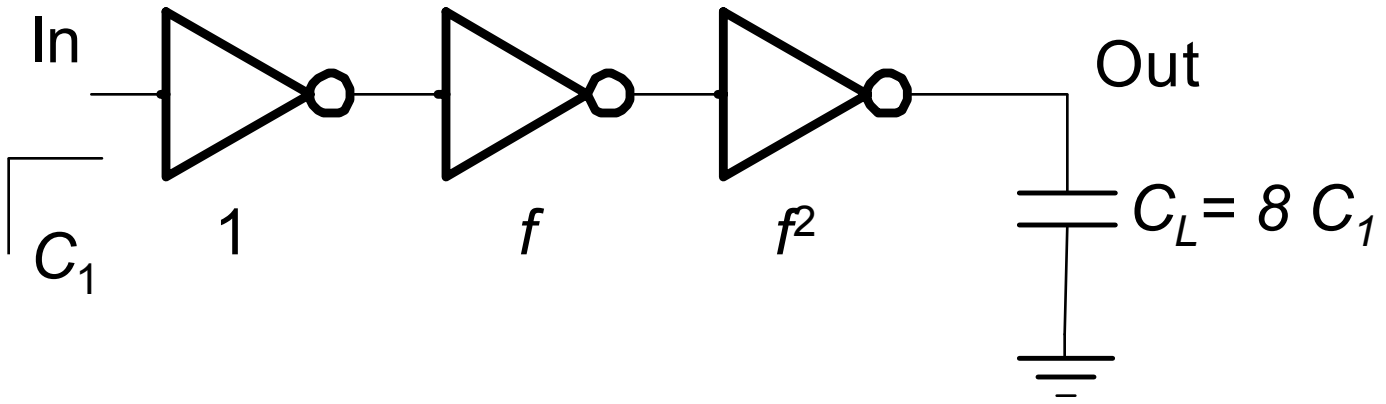
$$t_p = \sum_{j=1}^{N} t_{p,j} = t_{p0} \sum_{i=1}^{N} \left( 1 + \frac{C_{gin,j+1}}{\gamma C_{gin,j}} \right), \quad C_{gin,N+1} = C_L$$

Optimum: When each stage is sized by *f* and has same eff. fanout *f*:

$$f^N = F = C_L / C_{gin,1}$$

$$f = \sqrt[N]{F}$$

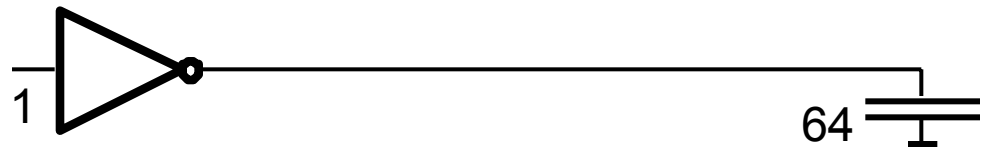
# Example



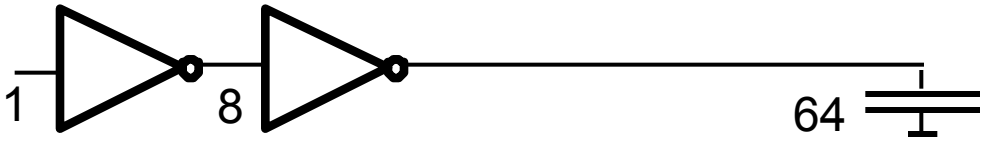

$C_L/C_1$ has to be evenly distributed across $N = 3$ stages:

$$f = \sqrt[3]{8} = 2$$

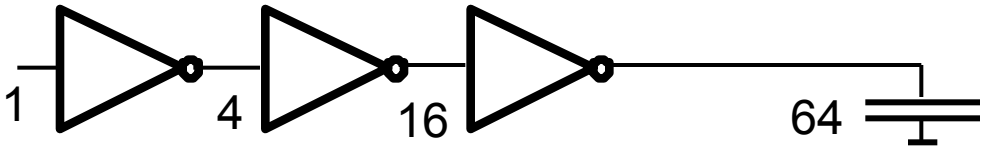
# Buffer Design



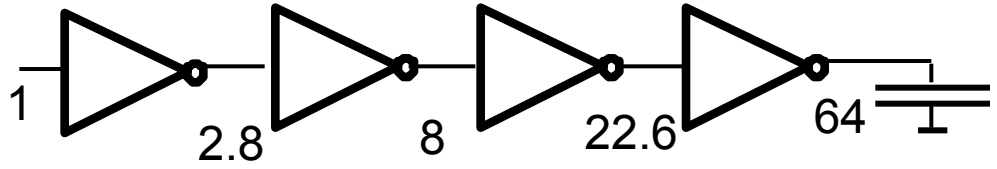

| N | f | $t_p$ |
|---|---|---|
| 1 | 64 | 65 |




| N | f | $t_p$ |
|---|---|---|
| 2 | 8 | 18 |



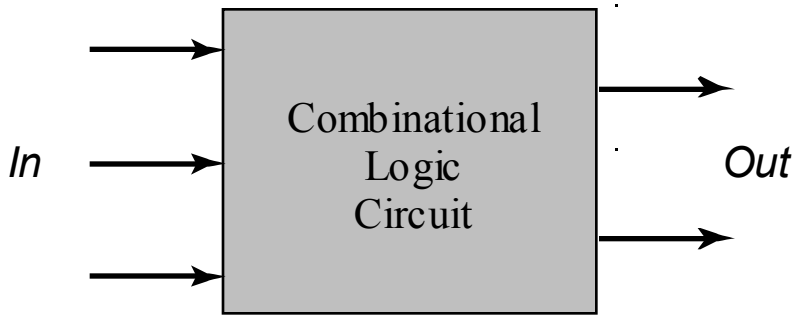

| N | f | $t_p$ |
|---|---|---|
| 3 | 4 | 15 |




| N | f | $t_p$ |
|---|---|---|
| 4 | 2.8 | 15.3 |

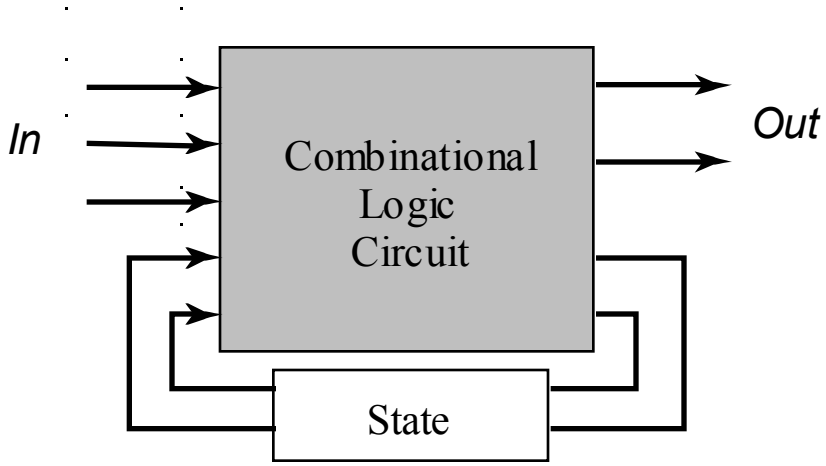
# Lógica Combinacional

# Combinational vs. Sequential Logic




Combinational

**Output = *f*(*In*)**




Sequential

**Output = *f*(*In*, *Previous In*)**

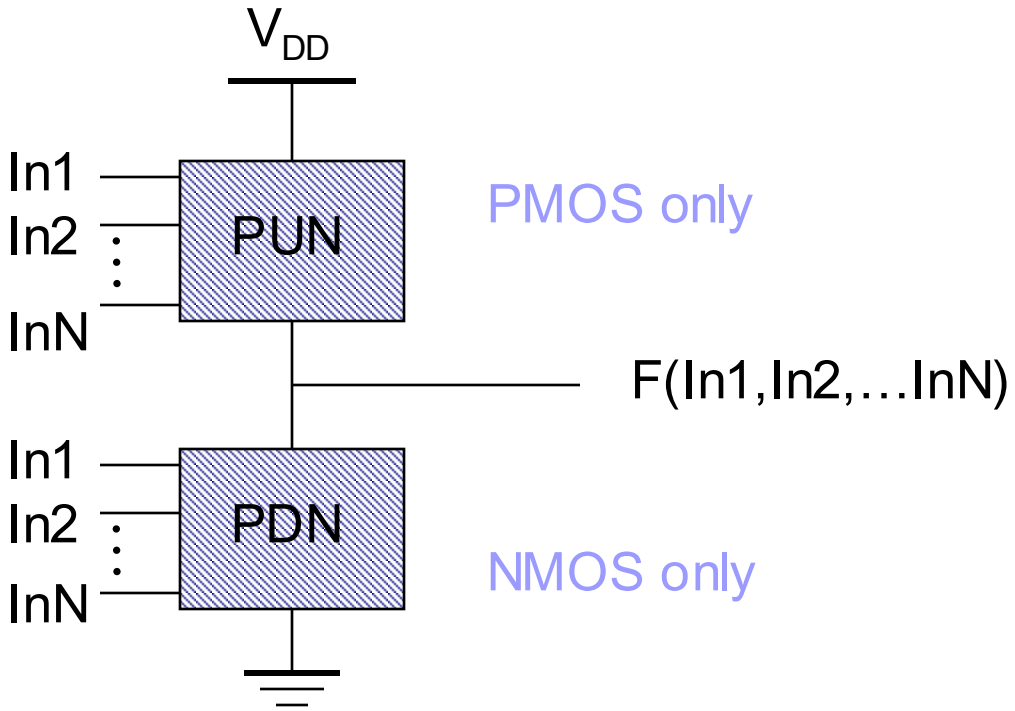
## Static CMOS Circuit

At every point in time (except during the switching transients) each gate output is connected to either $V_{DD}$ or $V_{ss}$ via a low-resistive path.

The outputs of the gates assume at all times the value of the Boolean function, implemented by the circuit (ignoring, once again, the transient effects during switching periods).

This is in contrast to the *dynamic* circuit class, which relies on temporary storage of signal values on the capacitance of high impedance circuit nodes.

# Static Complementary CMOS

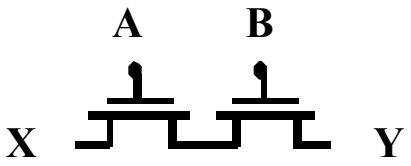



PUN and PDN are dual logic networks

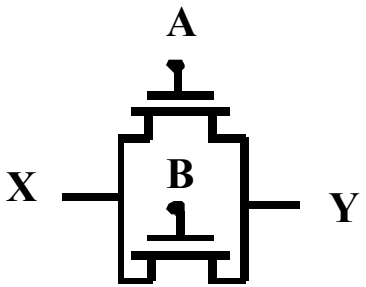
## NMOS Transistors in Series/Parallel Connection

**Transistors can be thought as a switch controlled by its gate signal**

**NMOS switch closes when switch control input is high**




**Y = X if A and B**

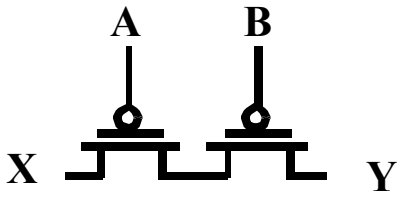



**Y = X if A OR B**

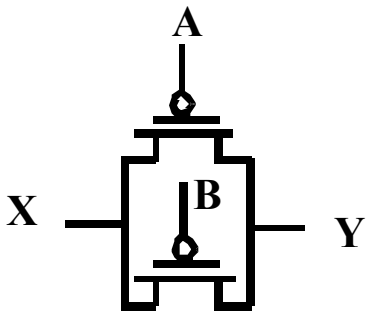
**NMOS Transistors pass a "strong" 0 but a "weak" 1**

## PMOS Transistors in Series/Parallel Connection

**PMOS switch closes when switch control input is low**




$Y = X$ if $\overline{A}$ AND $\overline{B} = \overline{A + B}$




$Y = X$ if $\overline{A}$ OR $\overline{B} = \overline{AB}$

**PMOS Transistors pass a "strong" 1 but a "weak" 0**

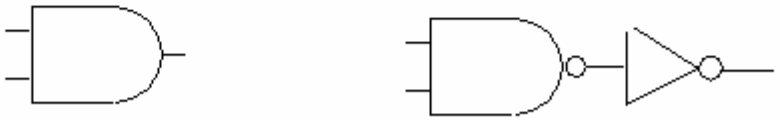
# Complementary CMOS Logic Style

- **PUP is the DUAL of PDN**
  **(can be shown using DeMorgan's Theorem's)**

$$\overline{A + B} = \bar{A}\bar{B}$$

$$\overline{AB} = \bar{A} + \bar{B}$$

- **The complementary gate is inverting**

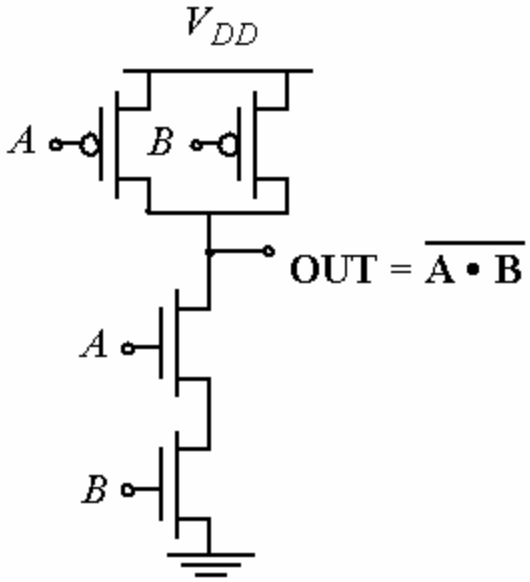


**AND = NAND + INV**

# Example Gate: NAND

| A | B | Out |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**Truth Table of a 2 input NAND gate**




**PDN: G = A B ⇒ Conduction to GND**

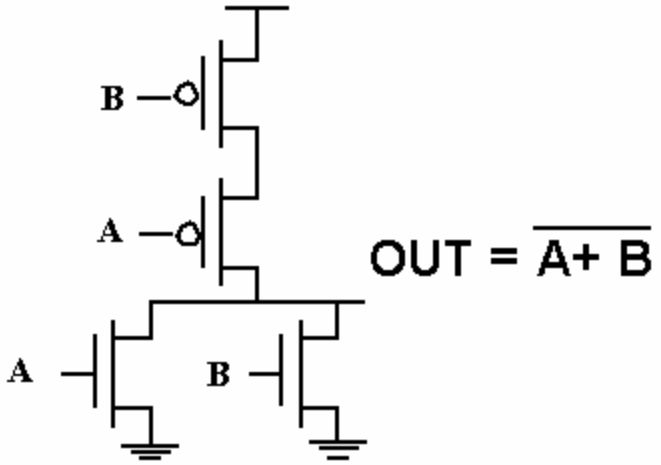
**PUN: F= $\overline{A} + \overline{B} = \overline{AB}$ ⇒ Conduction to $V_{DD}$**

$$\overline{G(In_1, In_2, In_3, \ldots)} \equiv F(\overline{In_1}, \overline{In_2}, \overline{In_3}, \ldots)$$

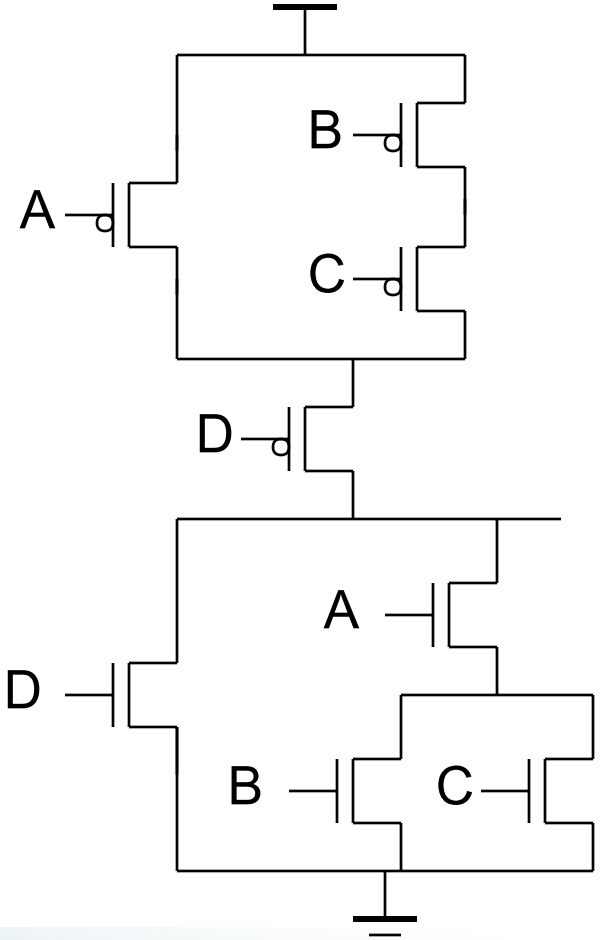
# Example Gate: NOR

| A | B | Out |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

Truth Table of a 2 input NOR gate

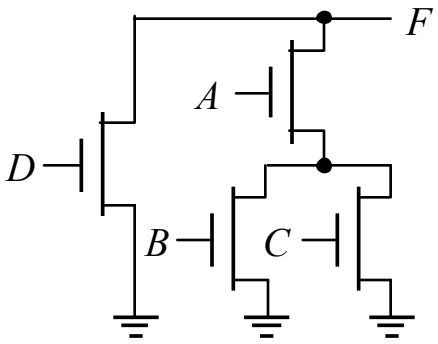


# Complex CMOS Gate

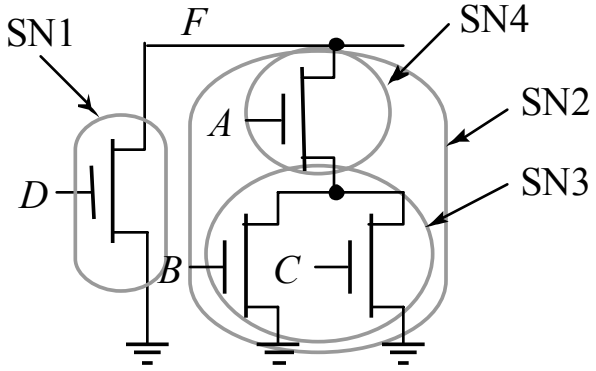



$$OUT = \overline{D + A \bullet (B + C)}$$

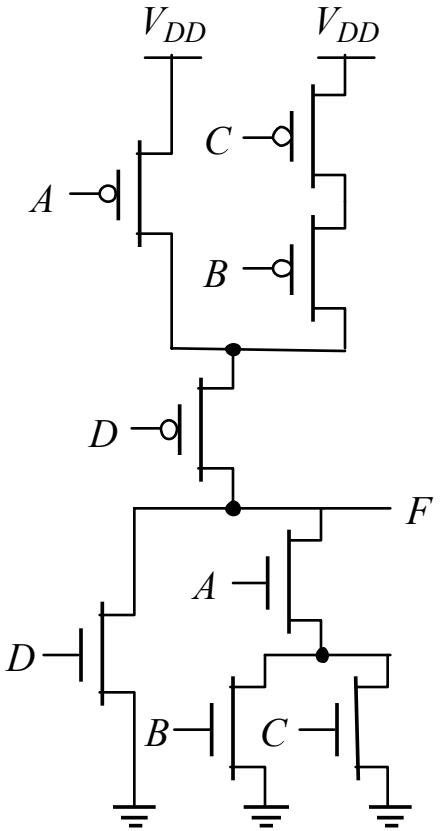
# Constructing a Complex Gate




(a) pull-down network




(b) Deriving the pull-up network hierarchically by identifying sub-nets

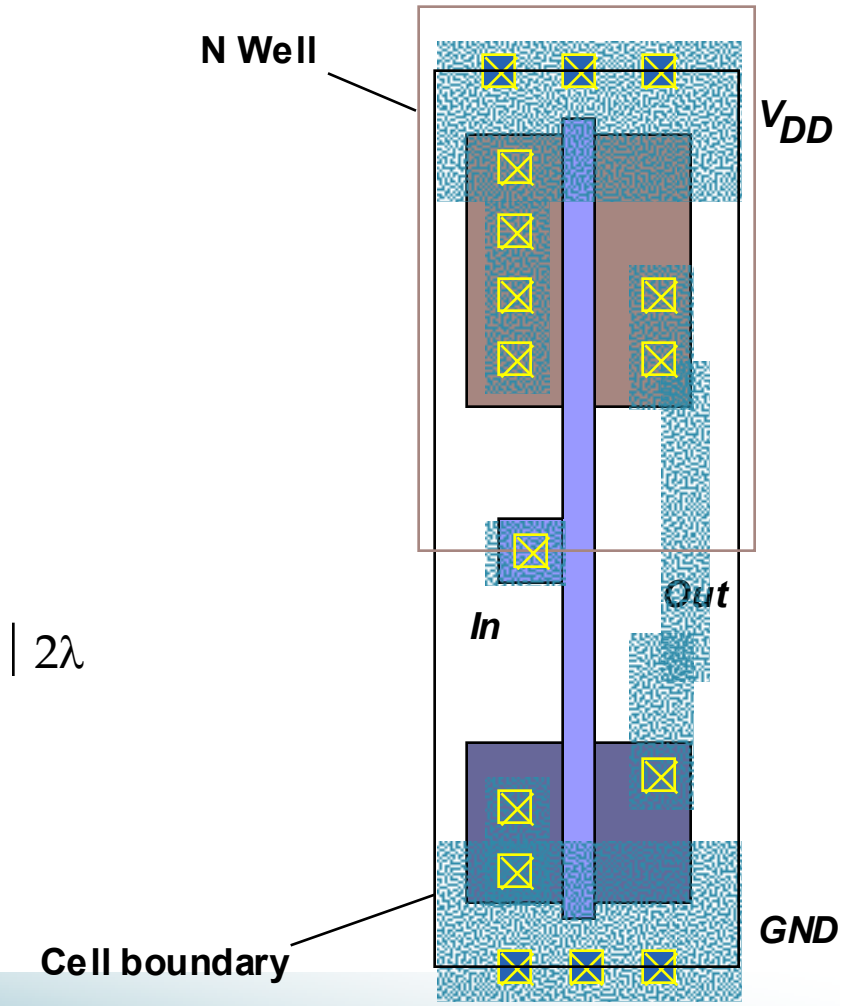



(c) complete gate

# Cell Design

- Standard Cells
  - General purpose logic
  - Can be synthesized
  - Same height, varying width
- Datapath Cells
  - For regular, structured designs (arithmetic)
  - Includes some wiring in the cell
  - Fixed height and width

# Standard cells



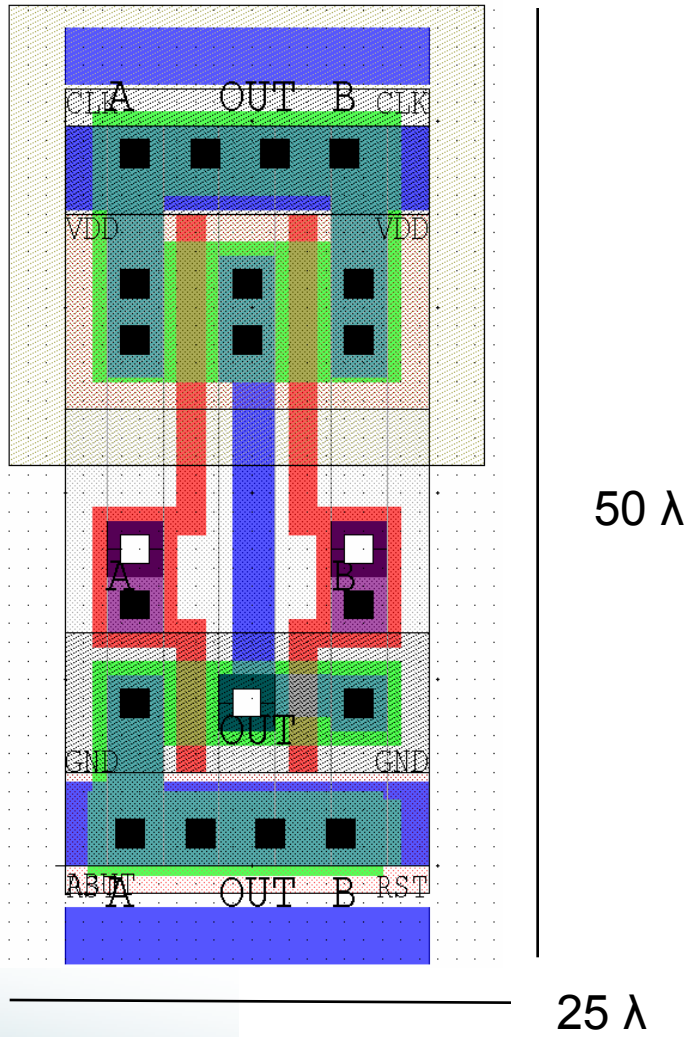

Cell height 12 metal tracks
Metal track is approx. 3λ + 3λ
Pitch =
repetitive distance between objects

Cell height is "12 pitch"

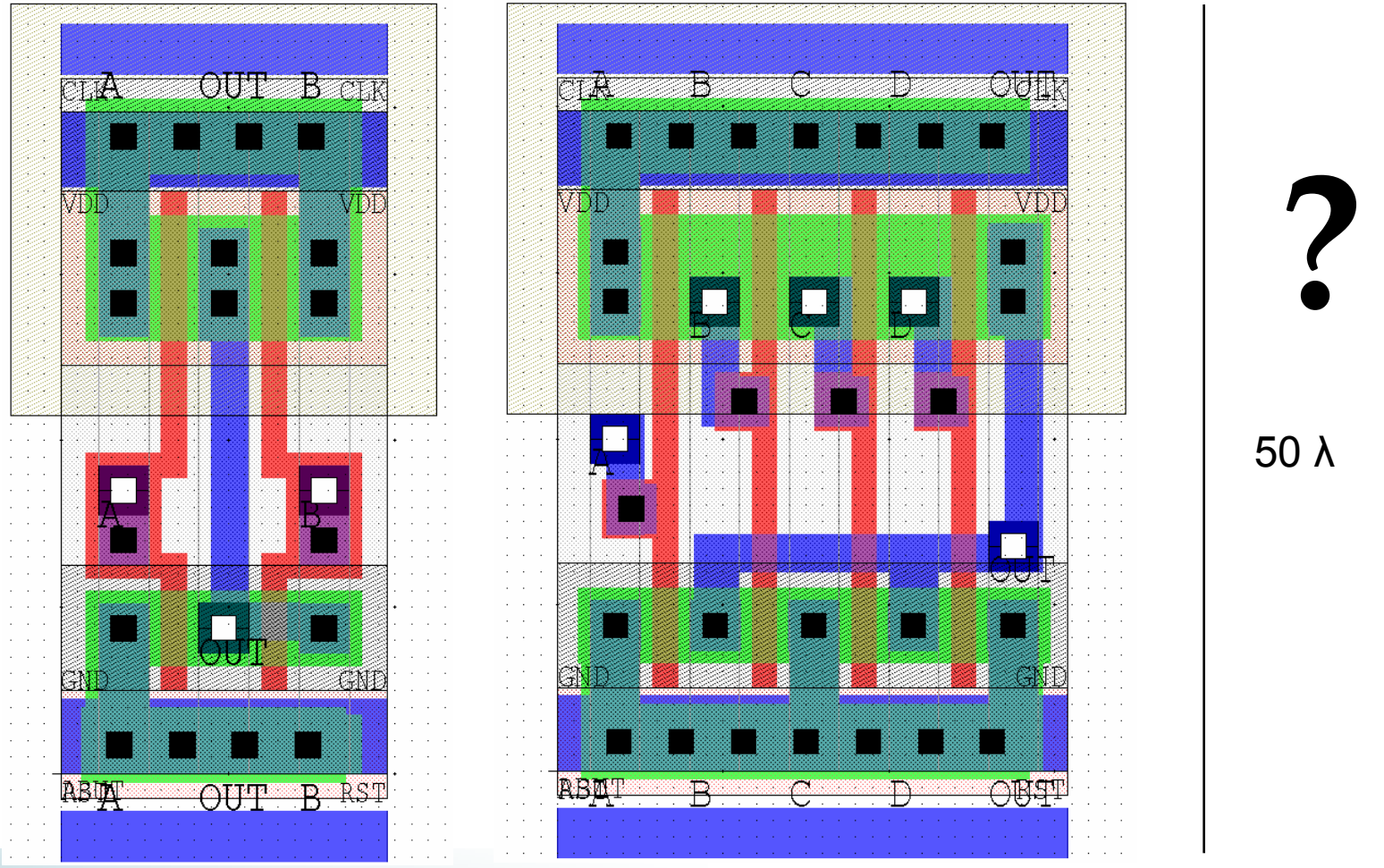
Rails ~10λ

# Standard cells

?



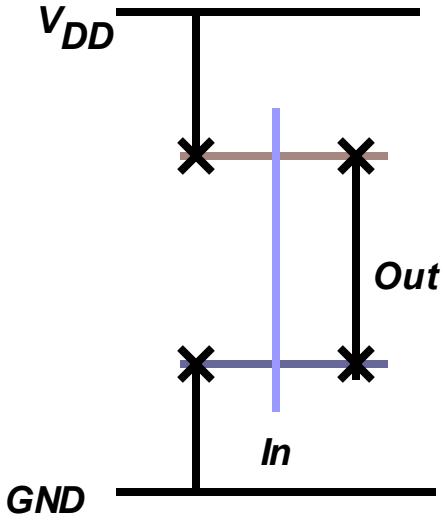
# Standard cells



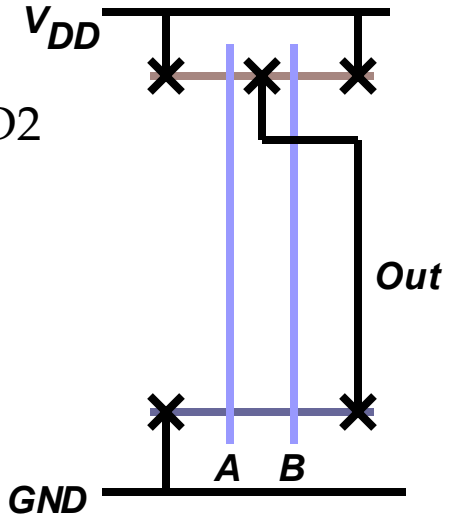
# Stick Diagrams

Contains no dimensions
Represents relative positions of transistors

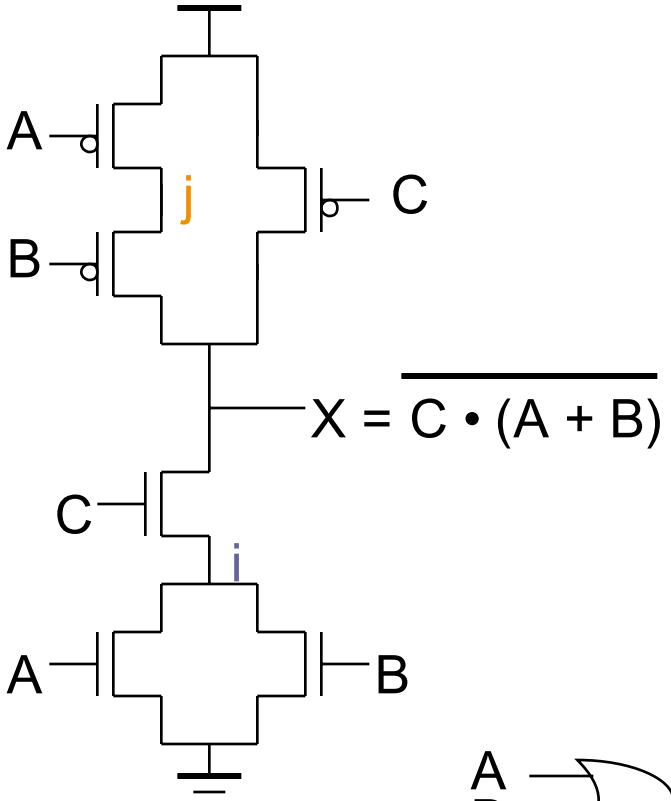
Inverter



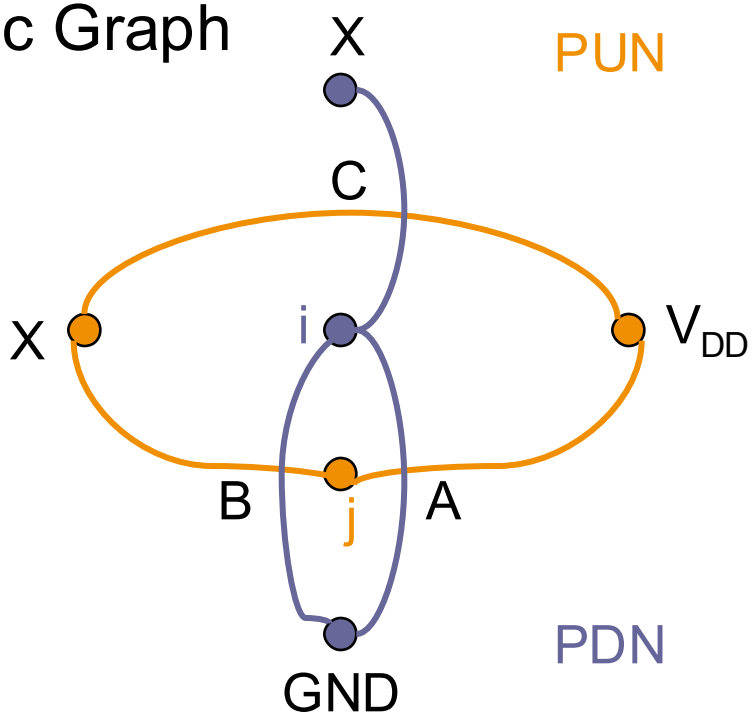

NAND2

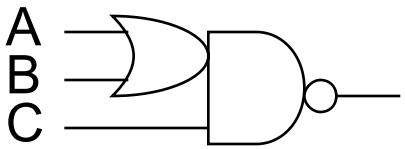


# Stick Diagrams

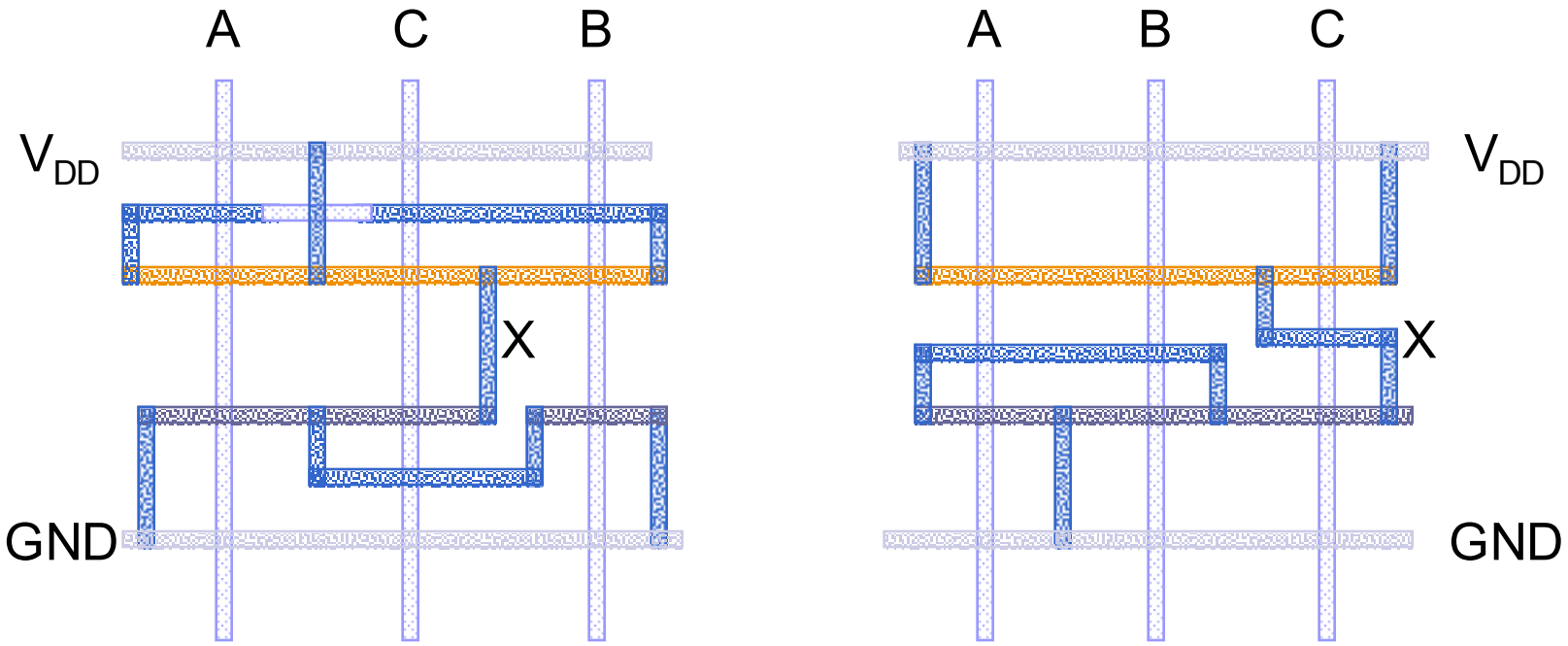



Logic Graph



*Two Versions of C • (A + B)*



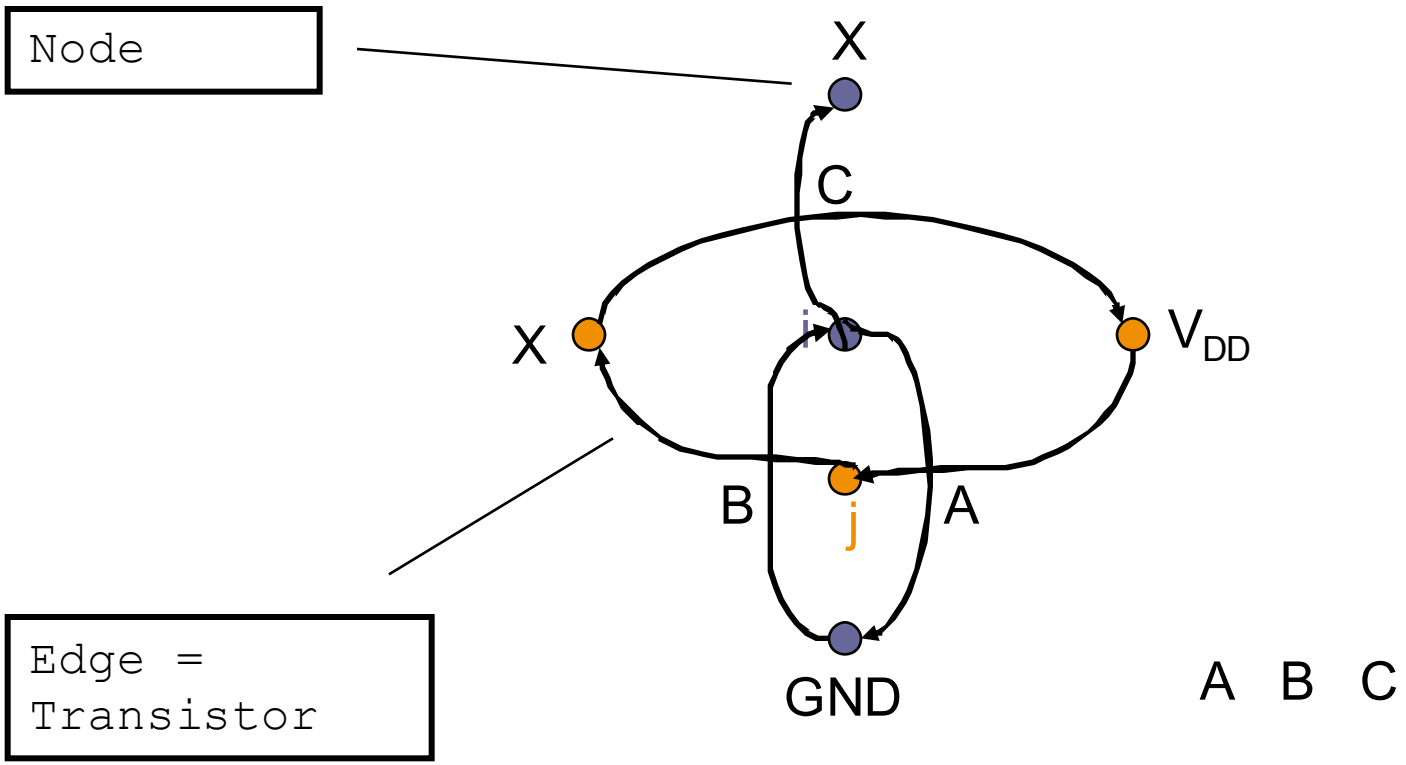

Permutation of input signals that produce uninterrupted active strips is important !

## Euler Paths

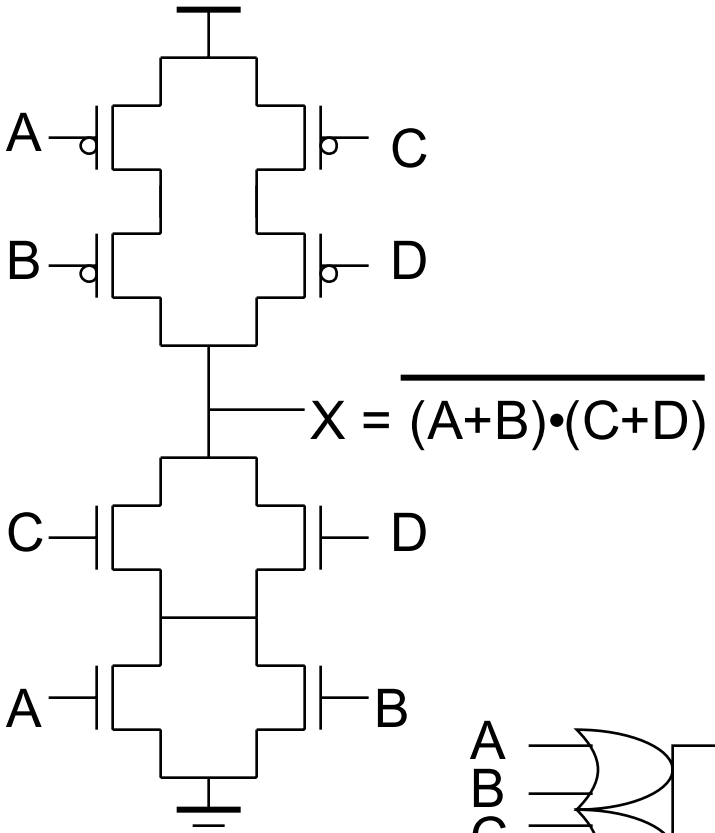
- There is a systematic approach to uninterrupted strips of active. Two steps:
- Step 1: Construction of logic graph
- Step 2: Identification of Euler graphs

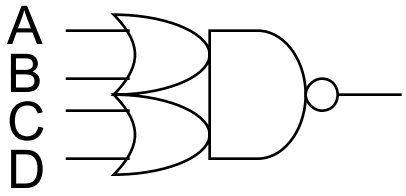
- Euler path is a path through all nodes such that every edge is visited once.
- Euler path is equivalent to an uninterrupted A-strip (succesive S and D connections)
- Consistency: Same ordering for PUN and PDN

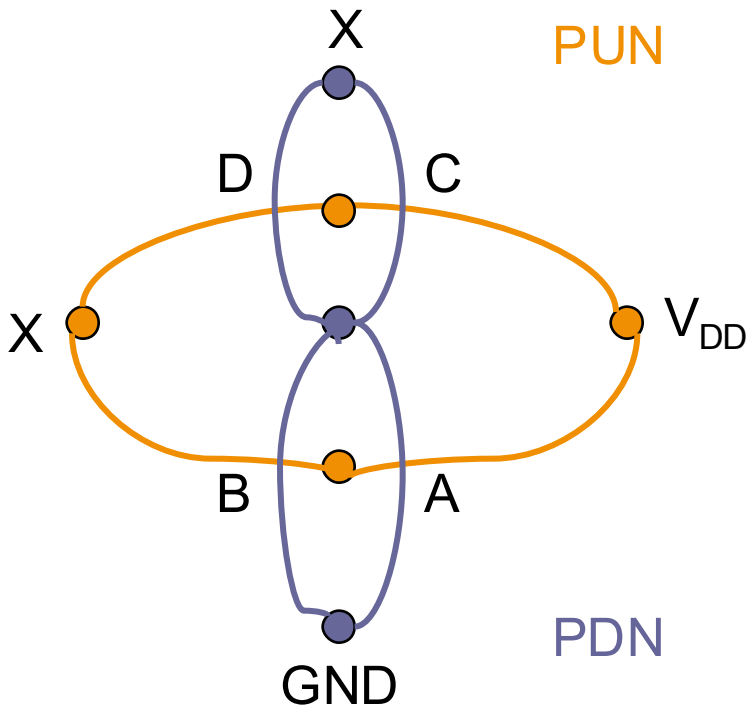
# Euler Path



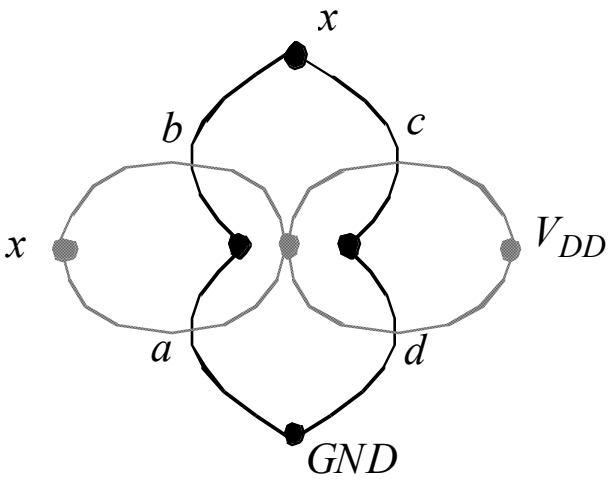
# OAI22 Logic Graph



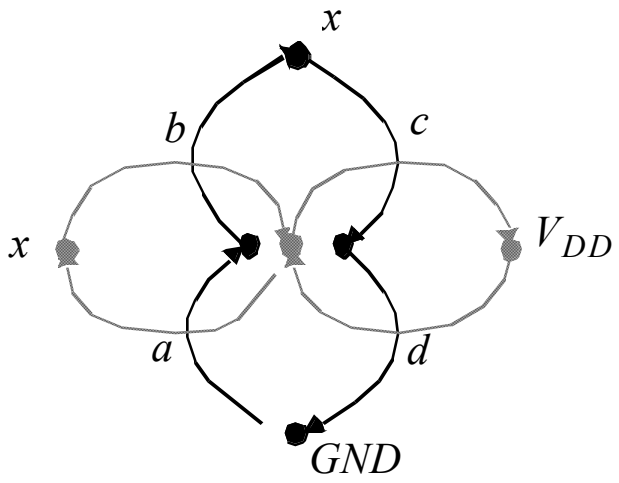



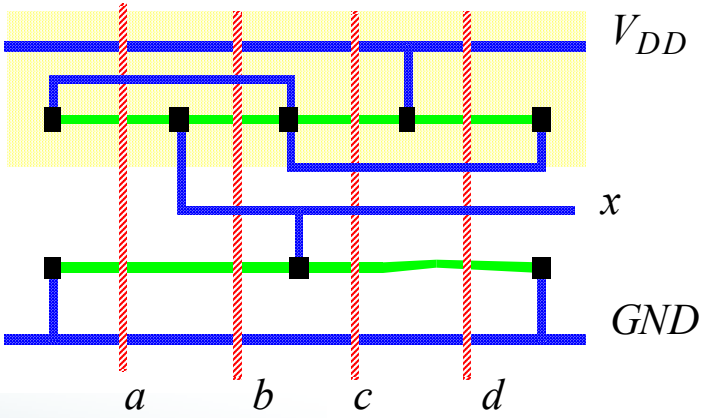
# Example: x = ab+cd




(a) Logic graphs for $\overline{(ab+cd)}$




(b) Euler Paths $\{a\ b\ c\ d\}$

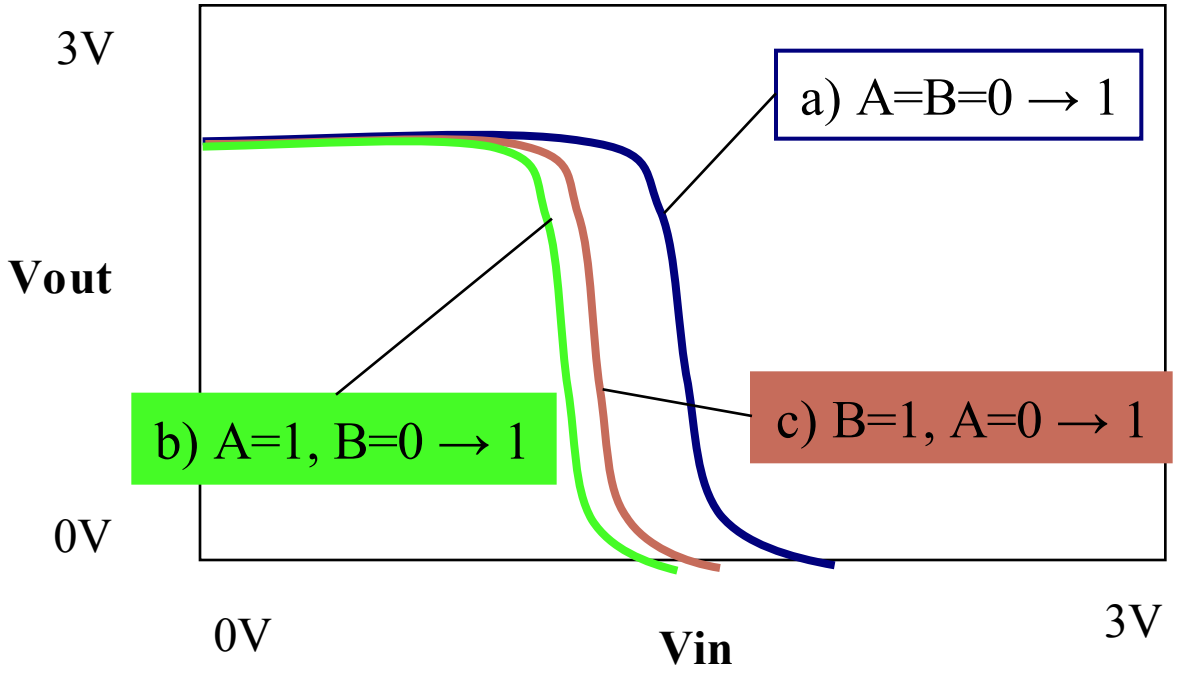


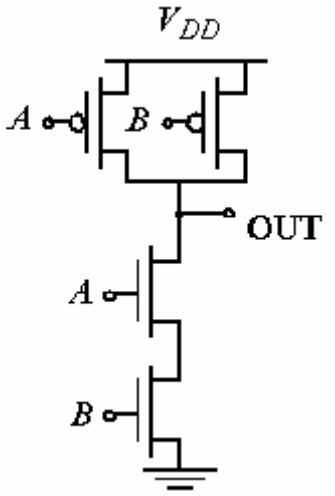

(c) stick diagram for ordering $\{a\ b\ c\ d\}$

# Properties of CMOS gates

# Static Properties

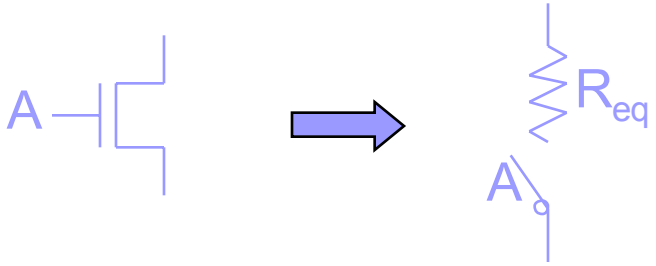
- Depend on input pattern

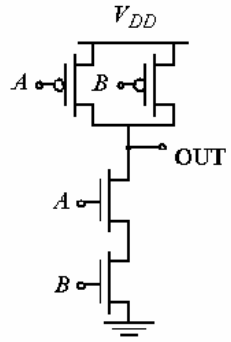


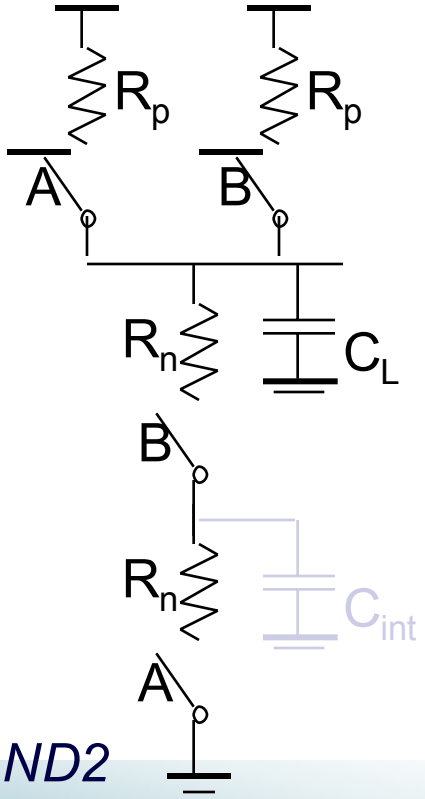




a) Two pull-up transistors in parallel are more difficult to turn off than one
b) One pull-up transistor, one pull-down. Dynamically, the internal node has to be discharge (slower)
c) Vds1 produces bulk effect during discharge. More Vin is needed

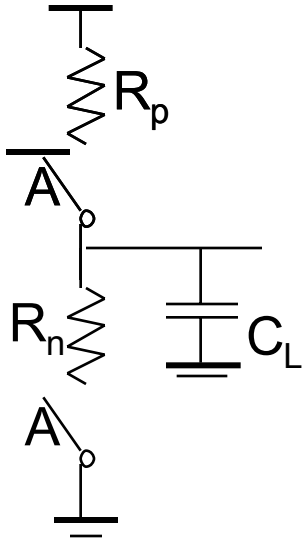
# Switch Delay Model



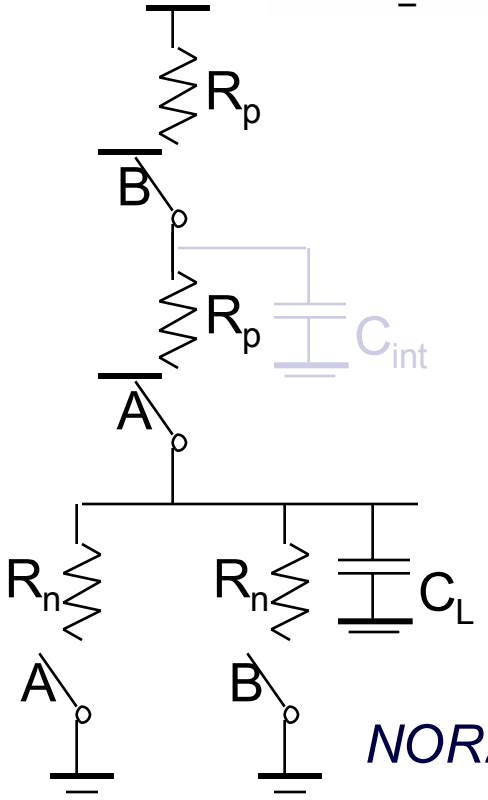




NAND2

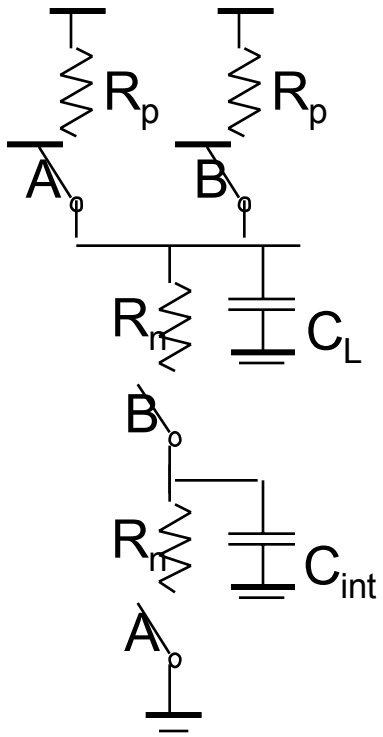



INV



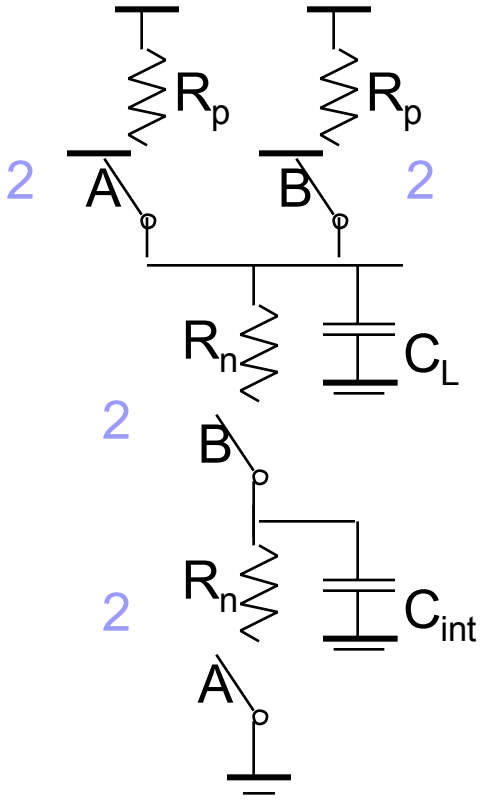

NOR2

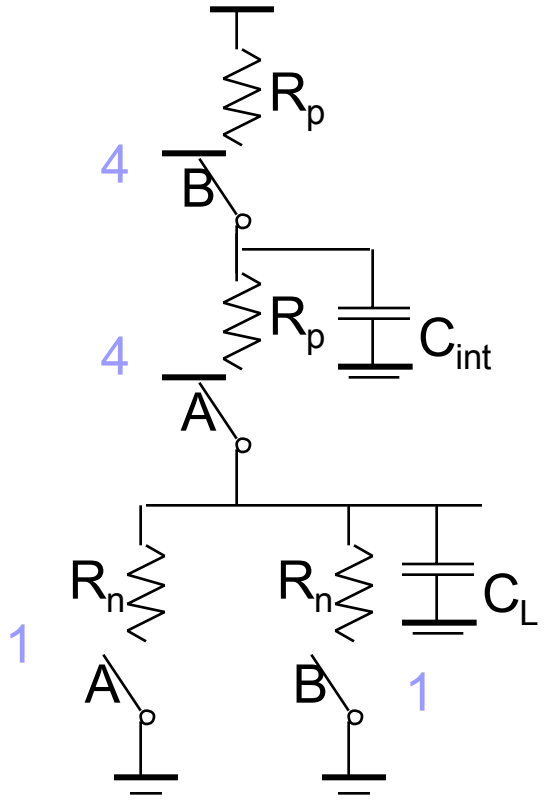
# Input Pattern Effects on Delay




- Delay is dependent on the pattern of inputs
- Low to high transition
  - both inputs go low
    - delay is 0.69 $R_p/2$ $C_L$
  - one input goes low
    - delay is 0.69 $R_p$ $C_L$
  - when N transistor A goes off, internal node has to be charged
- High to low transition
  - both inputs go high
    - delay is 0.69 $2R_n$ $C_L$

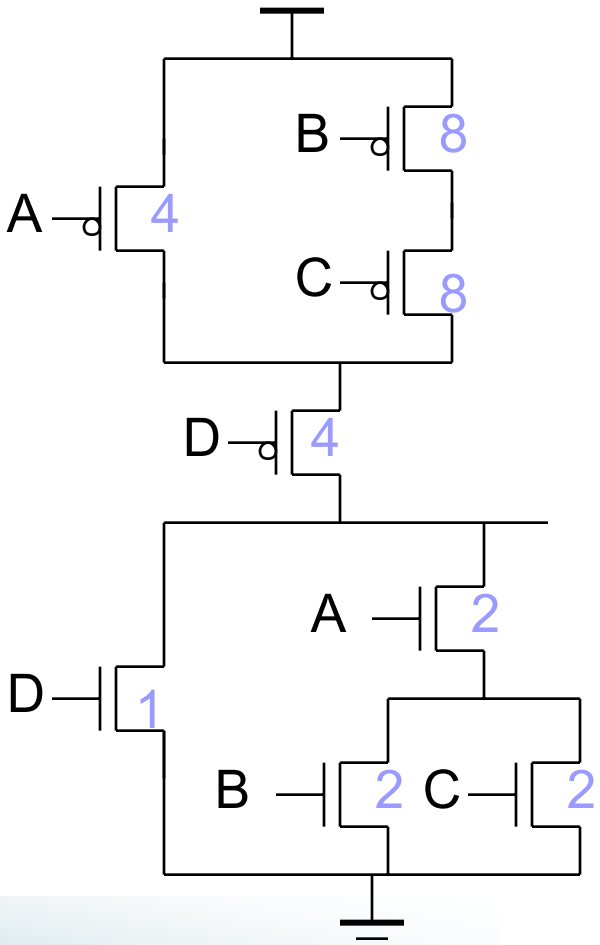
# Transistor Sizing



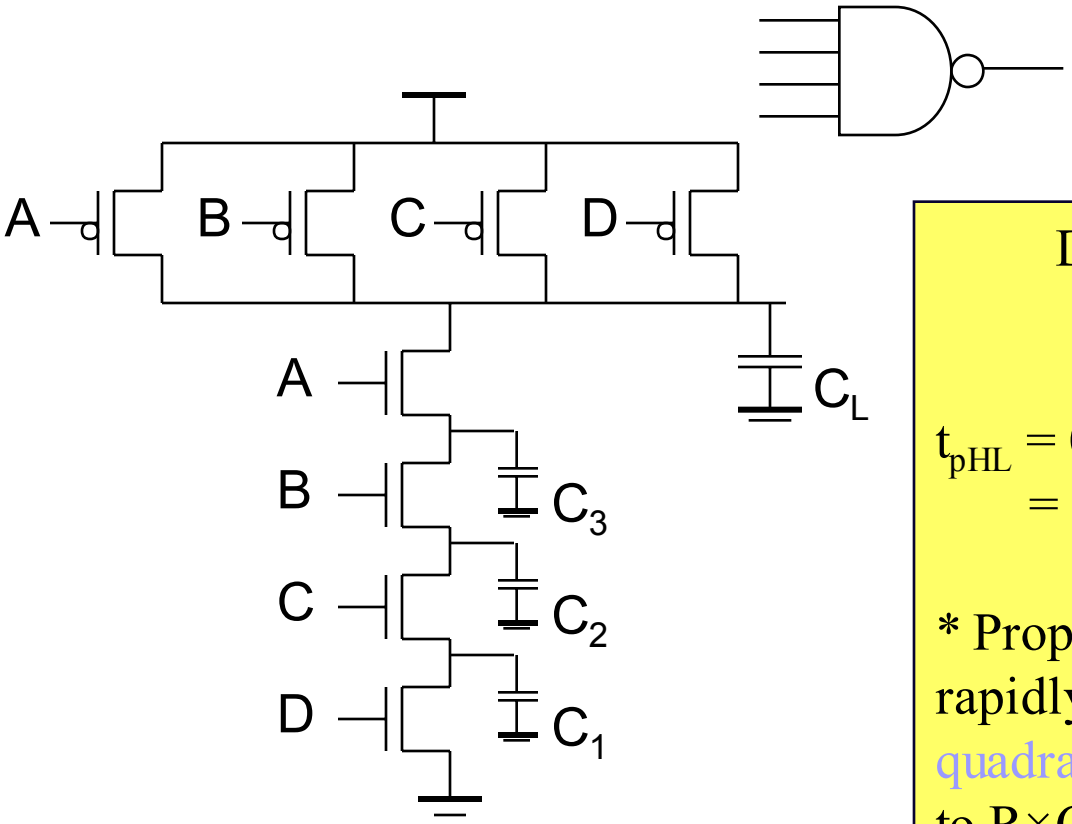

NAND based implementations are preferred over NOR …

# Transistor Sizing a Complex CMOS Gate




$$OUT = \overline{D + A \bullet (B + C)}$$

# Fan-In Considerations



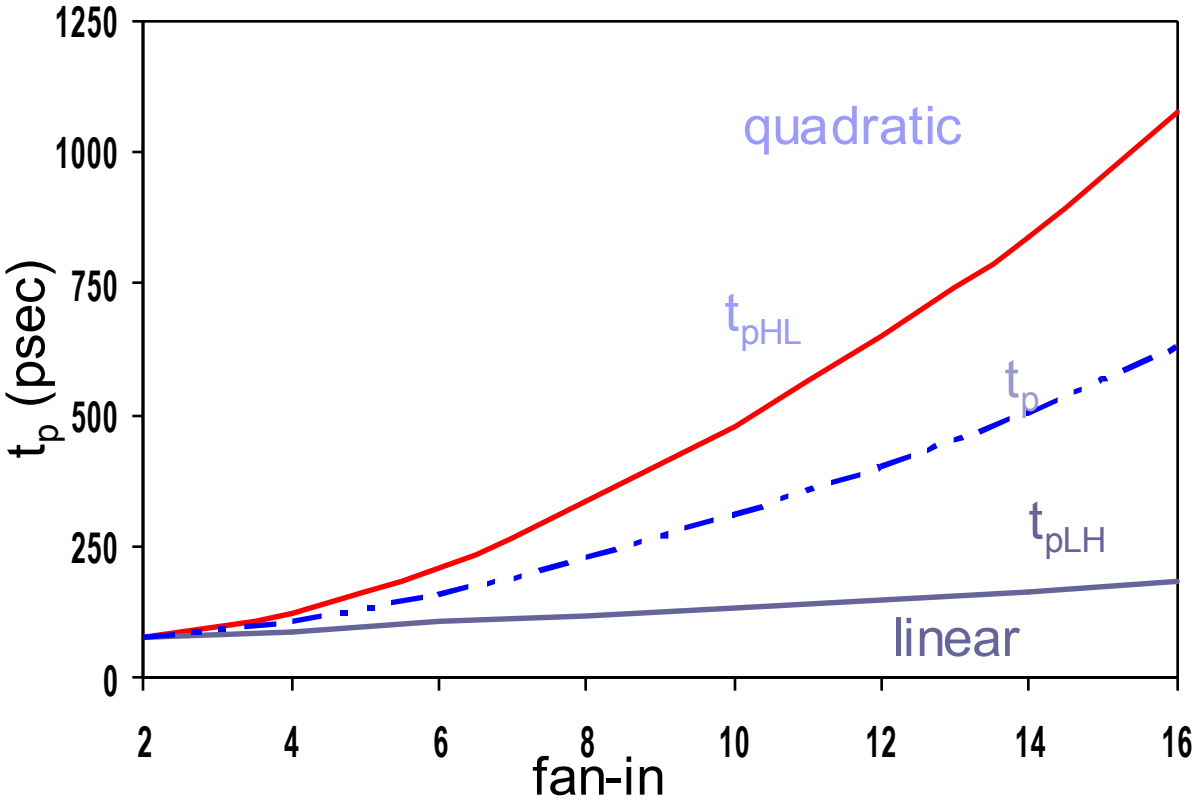

Distributed RC model
(Elmore delay)

$t_{pHL} = 0.69\ R_e\ (C_1+2C_2+3C_3+4C_L)$
$= R_e\ C_1+2\ R_e\ C_2+3R_e\ C_3+4R_e\ C_L$

* Propagation delay deteriorates rapidly as a function of fan-in – quadratically in the worst case. (prop. to R×C)

* Internal nodes important !!

# $t_p$ as a Function of Fan-In



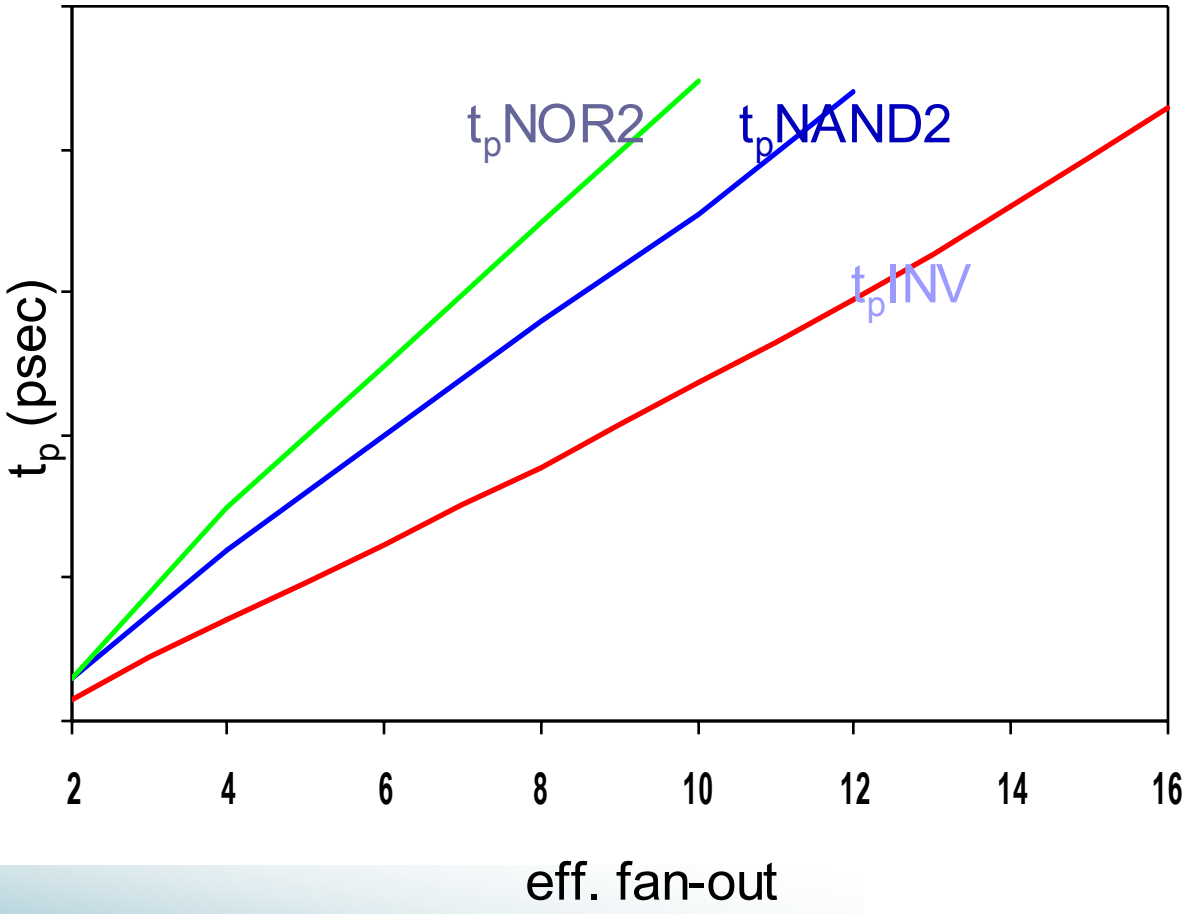

Intrinsec C increases linearly

Series transistors cause a double slowdown

Parallel transistors increase C

Gates with a fan-in greater than 4 should be avoided.

# $t_p$ as a Function of Fan-Out




All gates have the same drive current.

Slope is a function of "driving strength"

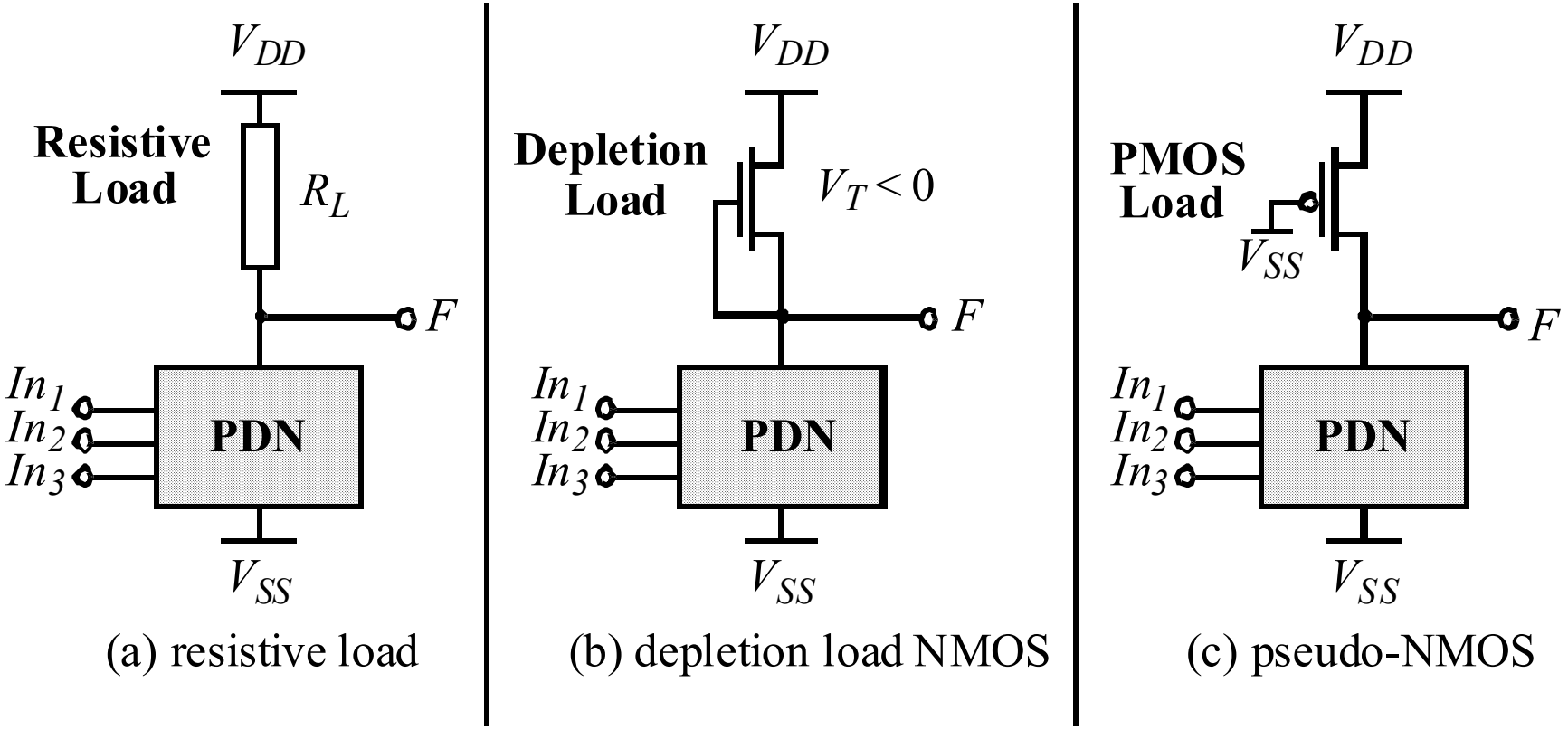
# $t_p$ as a Function of Fan-In and Fan-Out

- Fan-in: quadratic due to increasing resistance and capacitance
- Fan-out: each additional fan-out gate adds two gate capacitances to $C_L$

$$t_p = a_1 FI + a_2 FI^2 + a_3 FO$$

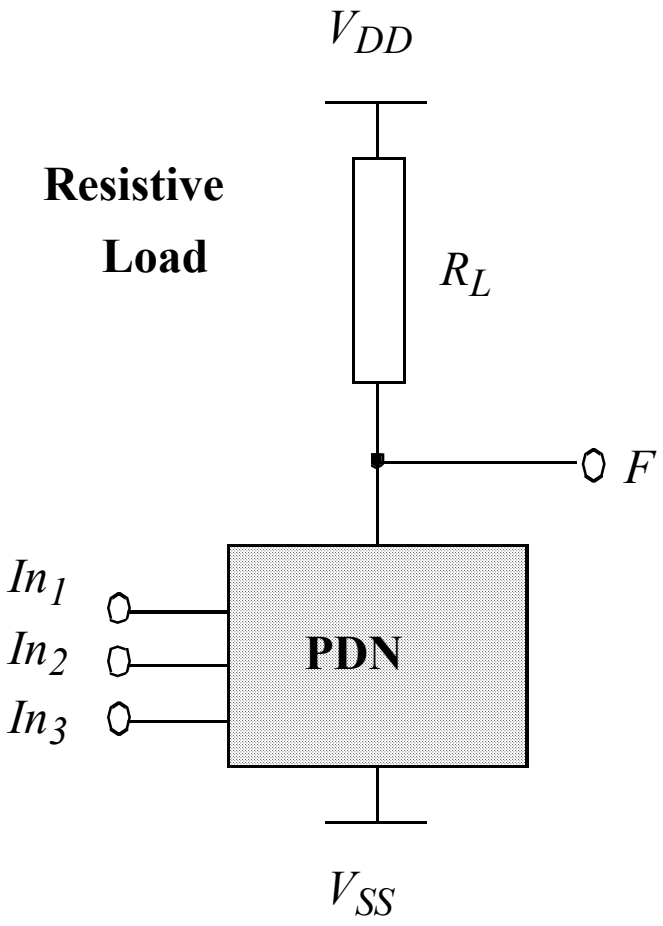
# Ratioed Logic

# Ratioed Logic



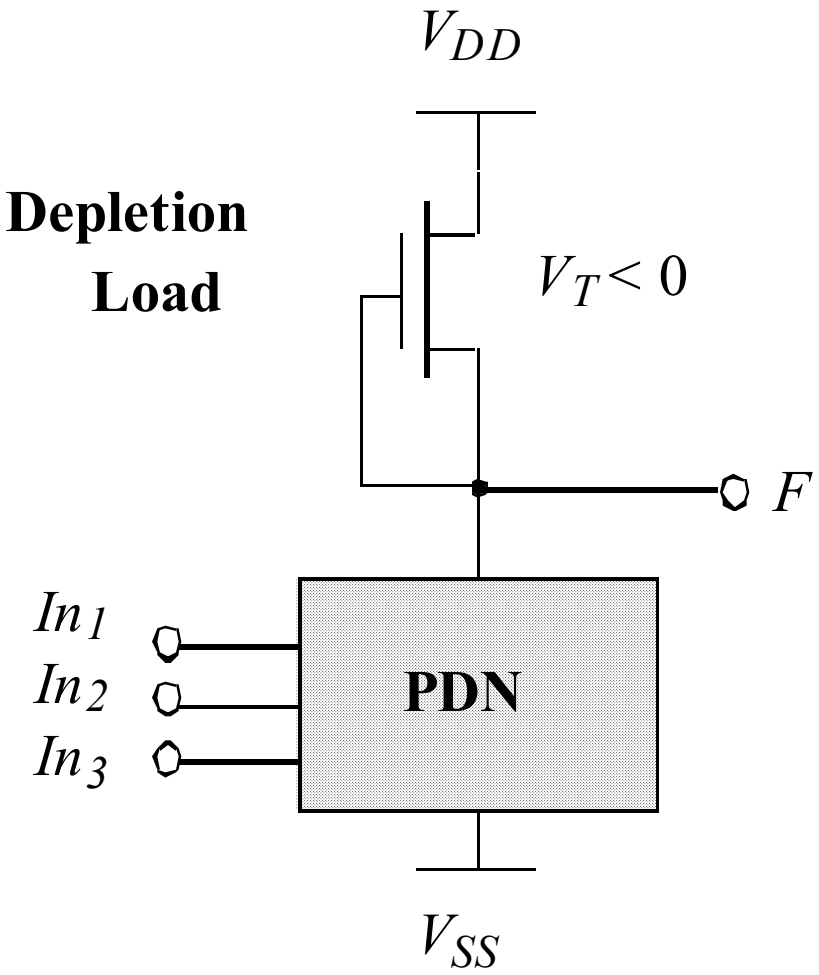

**Goal: to reduce the number of devices over complementary CMOS**

# Ratioed Logic

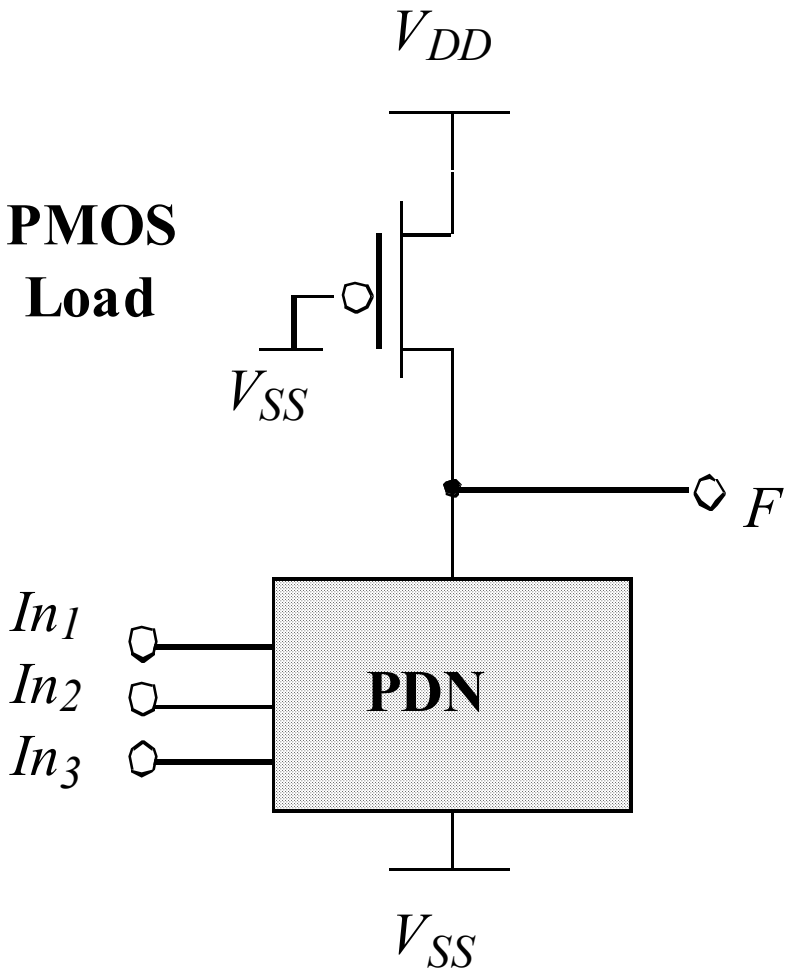



- N transistors + Load
- $V_{OH} = V_{DD}$
- $V_{OL} = \dfrac{R_{PN}}{R_{PN} + R_L}$
- Assymetrical response
- Static power consumption
- $t_{pL} = 0.69\ R_L C_L$

# Active Loads

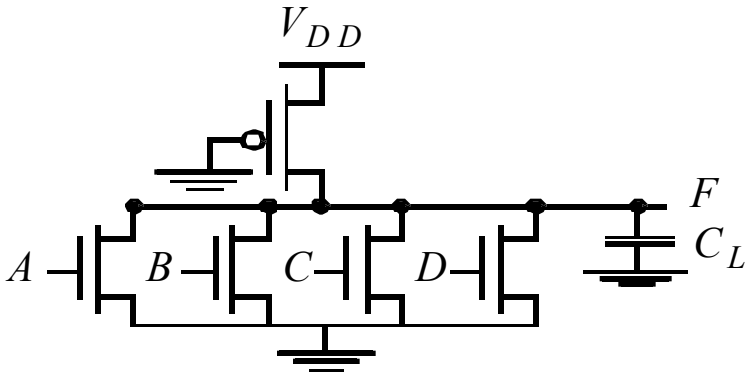



depletion load NMOS




pseudo-NMOS

## Pseudo-NMOS



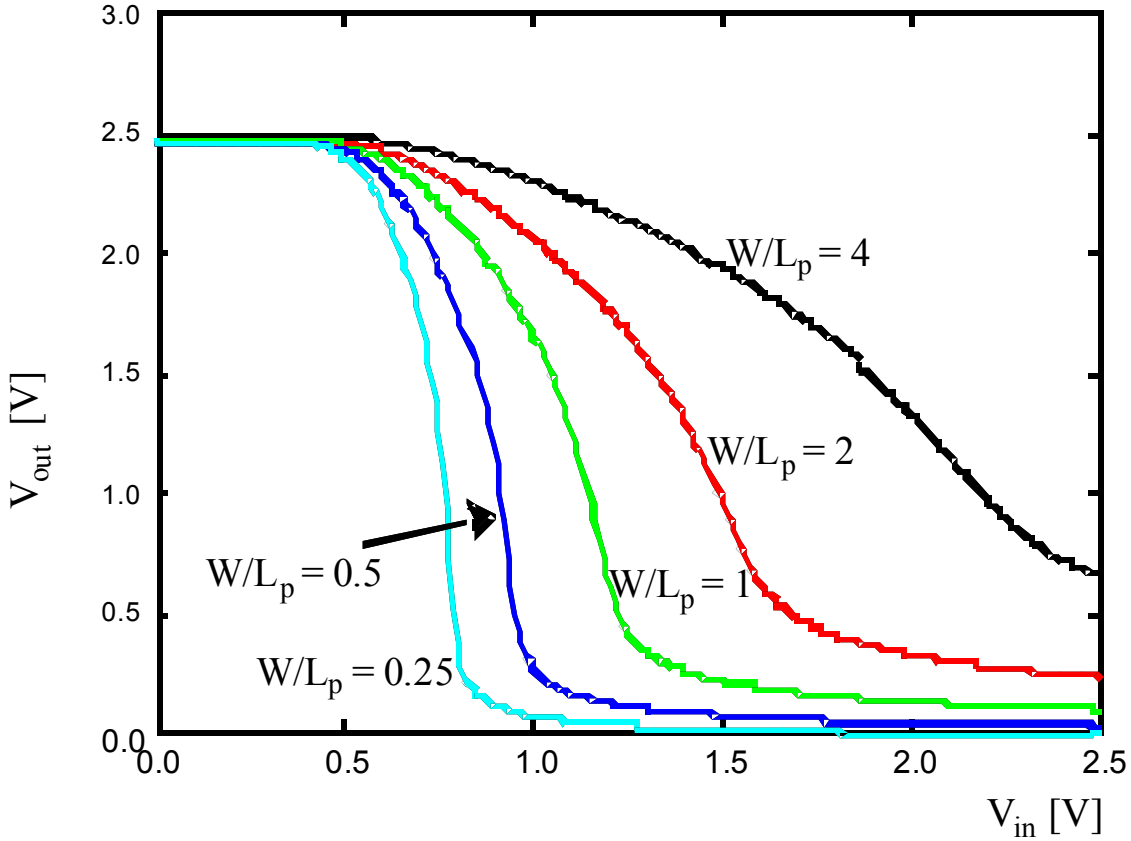

$$V_{OH} = V_{DD} \text{ (similar to complementary CMOS)}$$

$$k_n\left((V_{DD} - V_{Tn})V_{OL} - \frac{V_{OL}^2}{2}\right) = \frac{k_p}{2}(V_{DD} - |V_{Tp}|)^2$$

$$V_{OL} = (V_{DD} - V_T)\left[1 - \sqrt{1 - \frac{k_p}{k_n}}\right] \text{ (assuming that } V_T = V_{Tn} = |V_{Tp}|)$$

***SMALLER AREA & LOAD <u>BUT</u> STATIC POWER DISSIPATION!!!***

# Pseudo-NMOS VTC



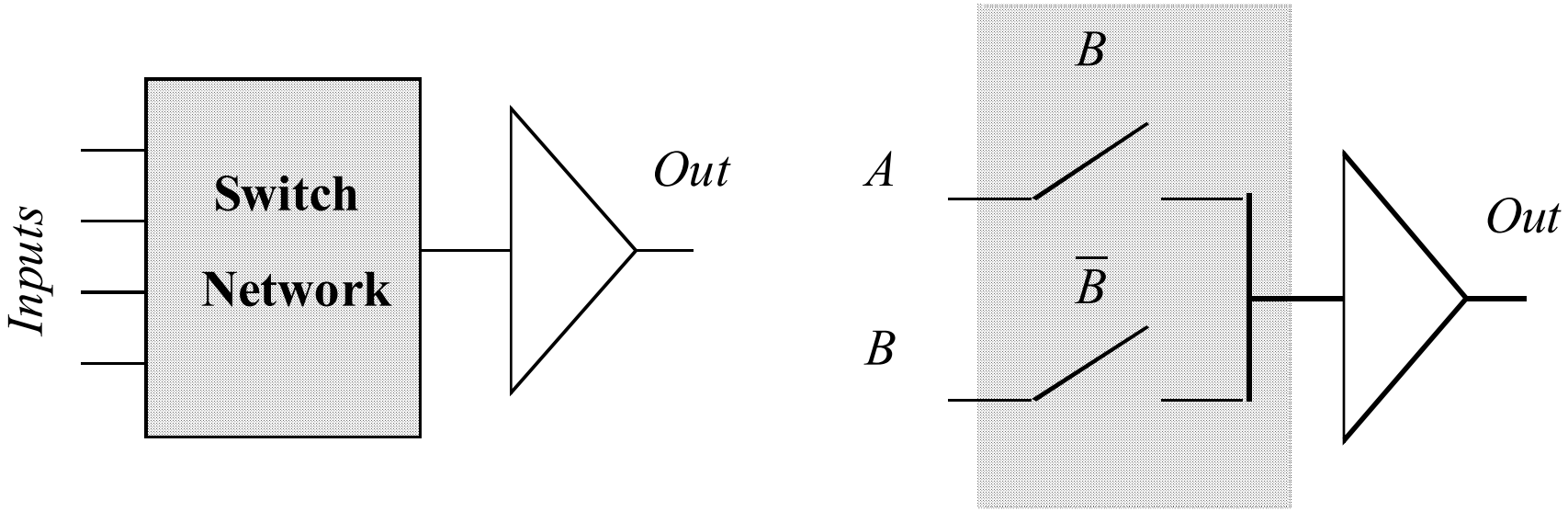

The bigger P, the faster the L to H transition, but more power and higher Vol

# Performance of a P-NMOS inverter

| Size | Vol | Static P | Tplh |
|---|---|---|---|
| 4 | 0.693 | 564µW | 14 ps |
| 2 | 0.273 | 298 µW | 56 ps |
| 1 | 0.133 | 160 µW | 123 ps |
| ½ | 0.064 | 80 µW | 268 ps |
| 1/4 | 0.031 | 41 µW | 569 ps |

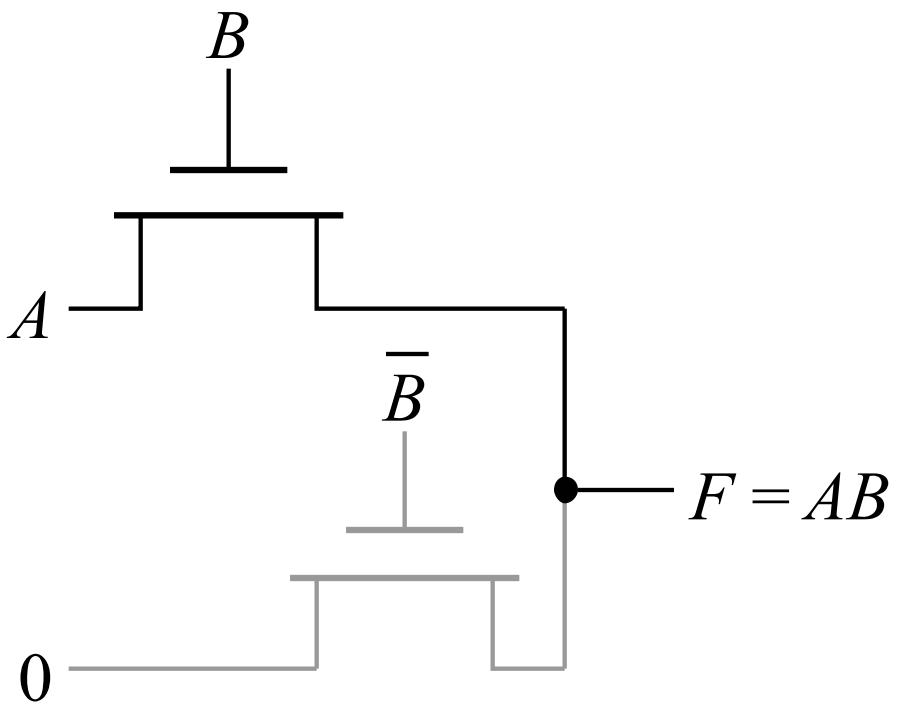
# Pass-Transistor Logic

# Pass-Transistor Logic

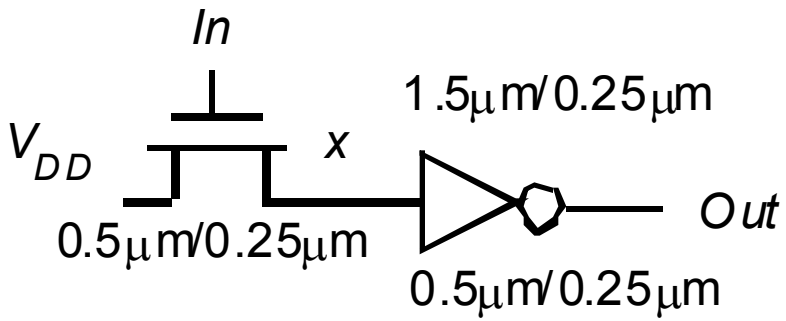


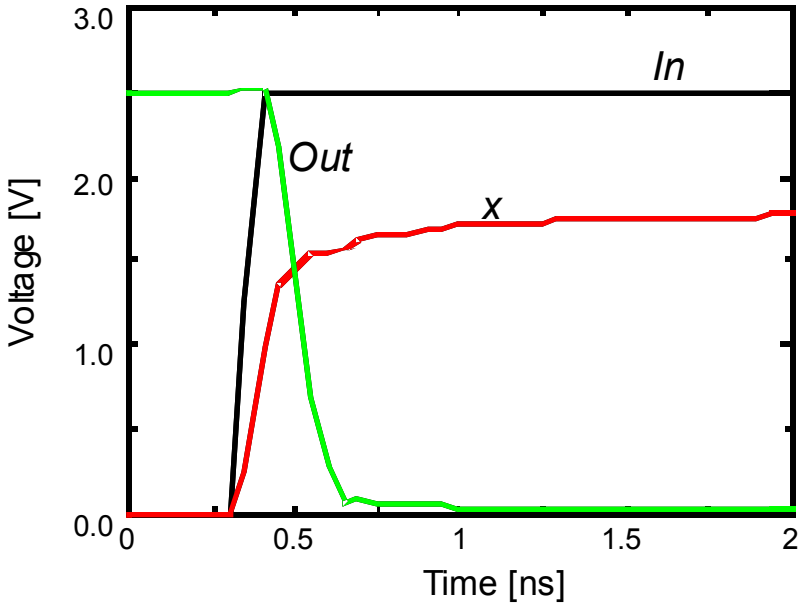

- N transistors
- No static consumption

# Example: AND Gate



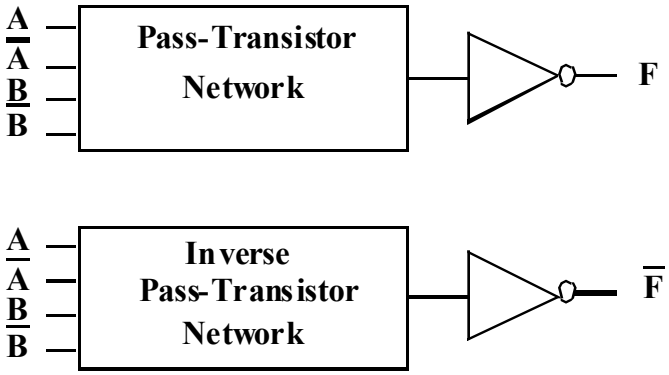
# NMOS-Only Logic



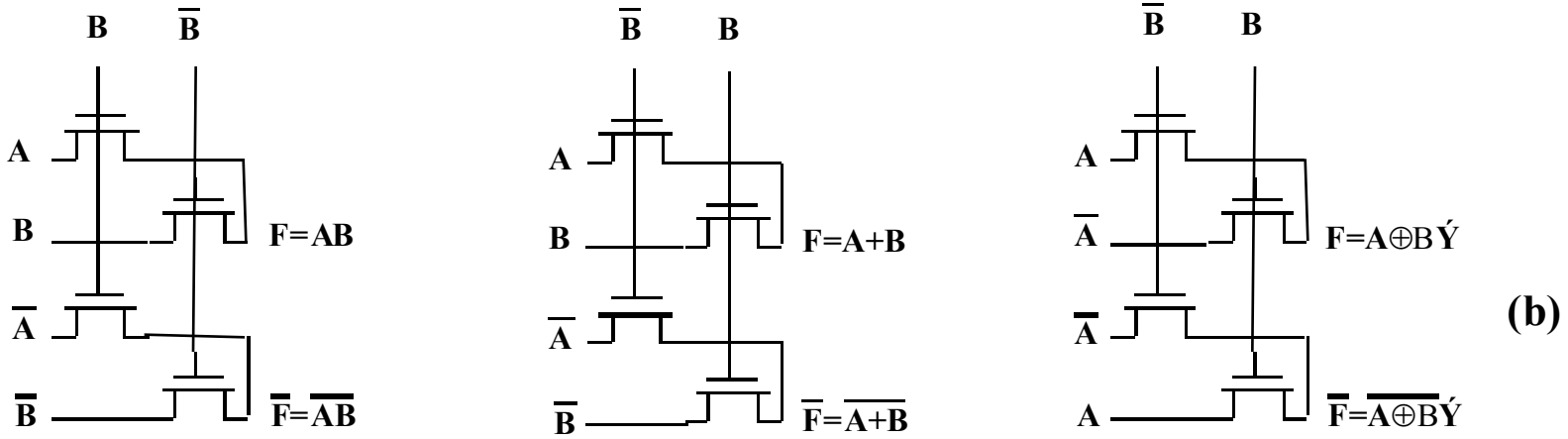




Tail end of transient very slow due to the small current available

# Complementary Pass Transistor Logic




(a)



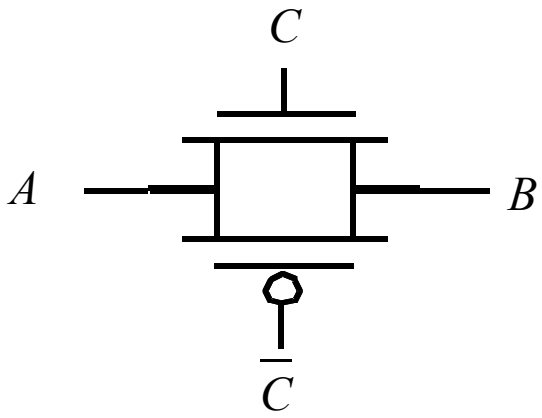

(b)

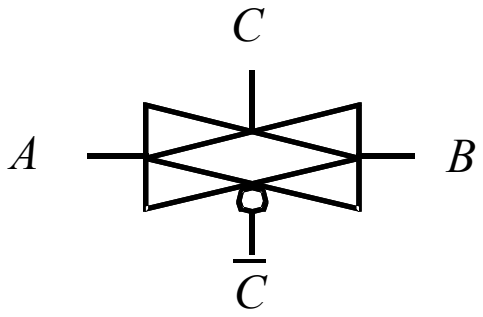
## Complementary Pass Transistor Logic

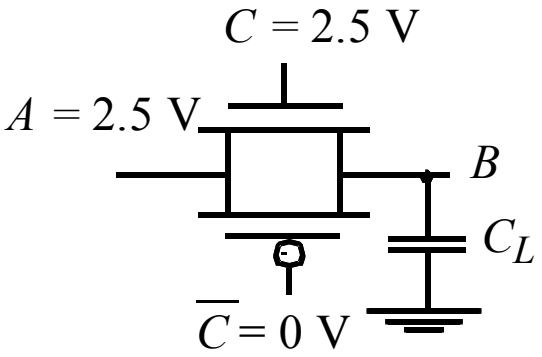
- Complementary data inputs and outputs are always available
- Output are always connected to low-impedance
- Design is very modular. The same topology is used. Permutation of inputs is used.

- Complementary signals have to be routed
- Restorer has to be used, otherwise static consumption

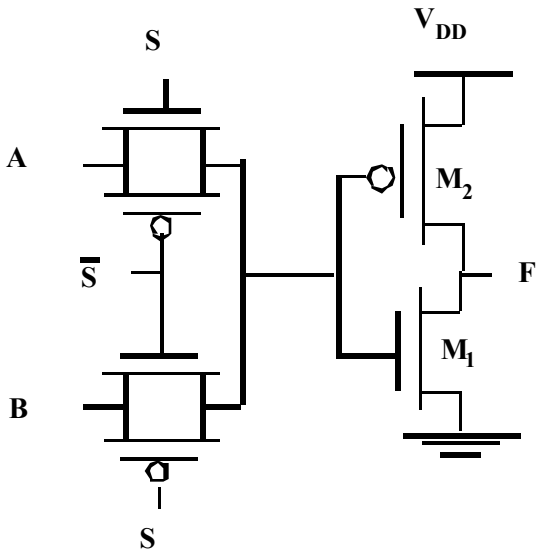
# Solution: Transmission Gate

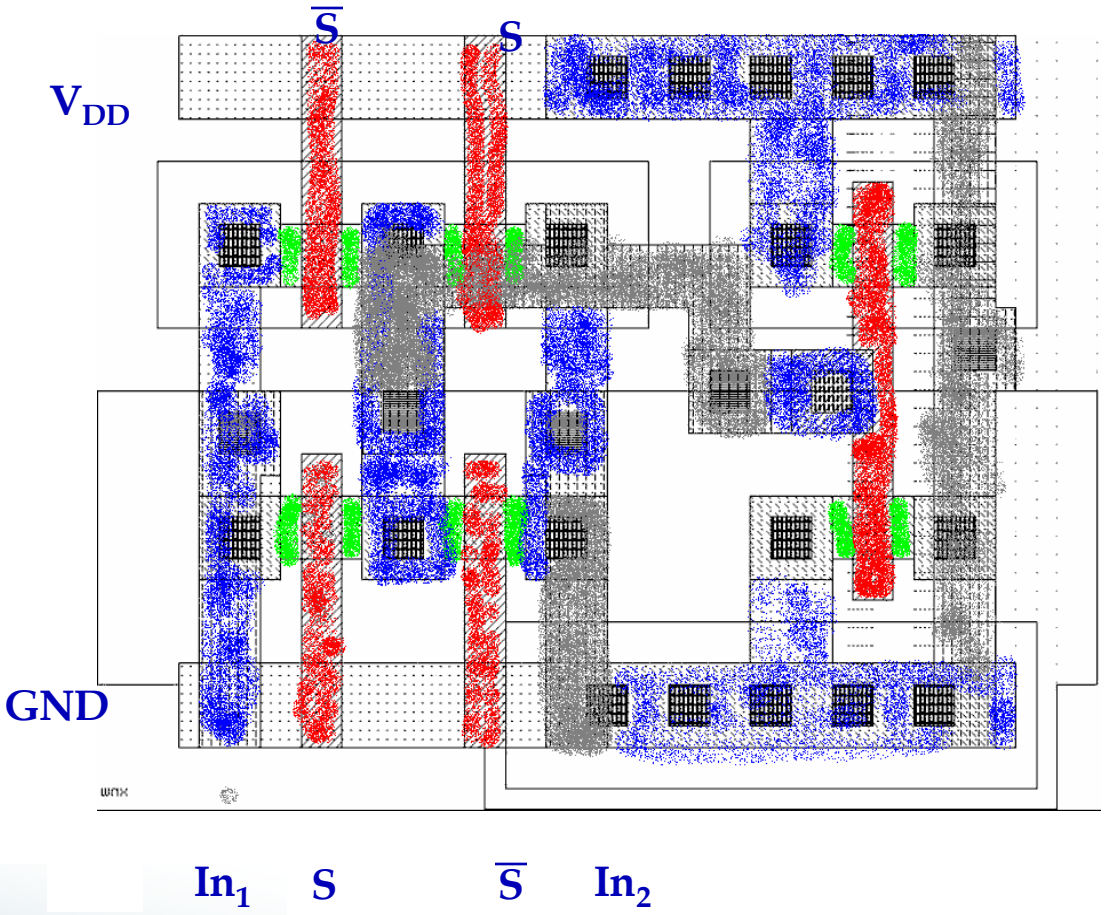


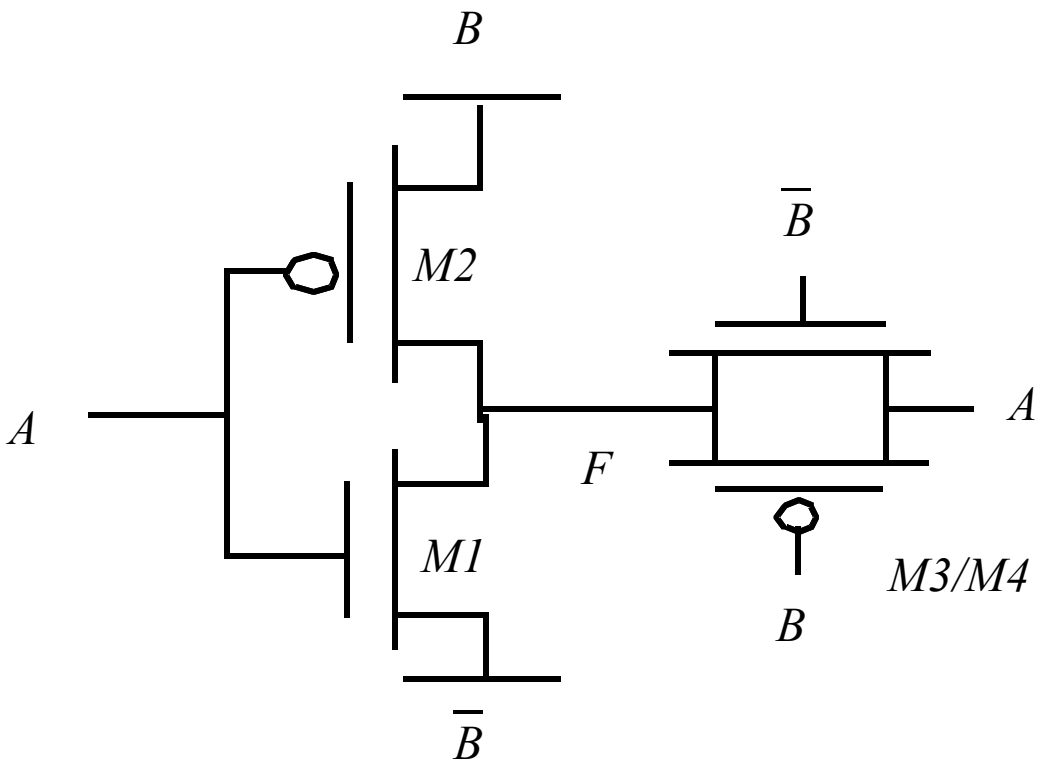



# Pass-Transistor Based Multiplexer

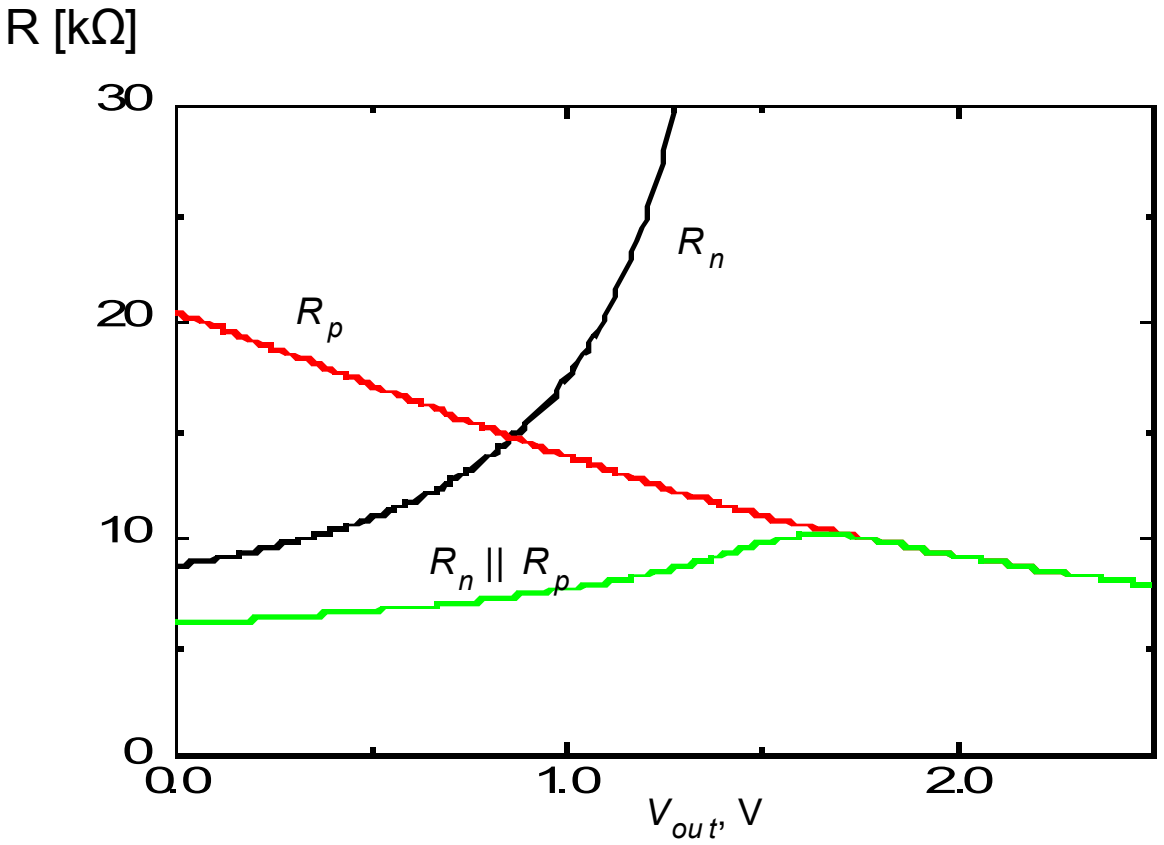


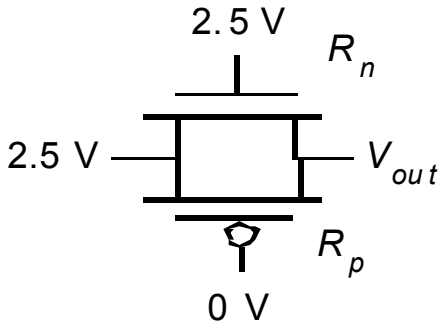



# Transmission Gate XOR



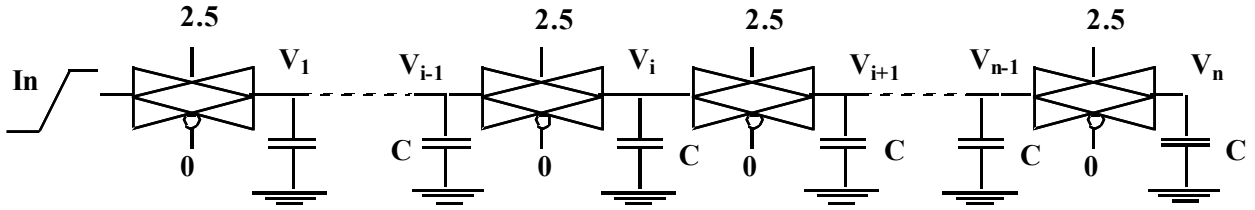
# Resistance of Transmission Gate



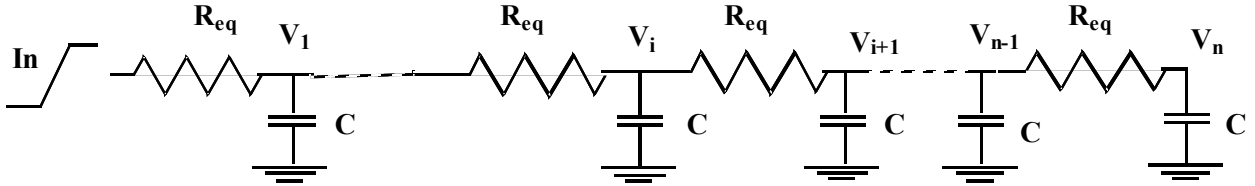




It has a series resistance that can be assumed constant, and equal to a couple of KΩ

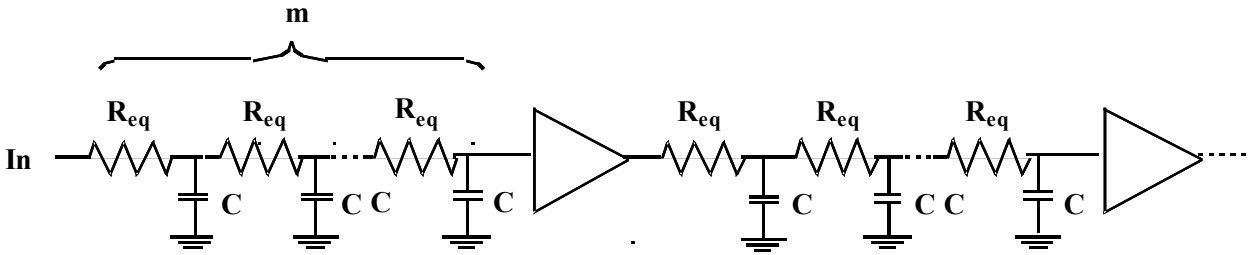
# Delay in Transmission Gate Networks




(a)




(b)




(c)

## Delay Optimization

- **Delay of RC chain**

$$t_p = 0.69 \sum_{k=0}^{n} CR_{eq} k = 0.69 CR_{eq} \frac{n(n+1)}{2}$$

- **Delay of Buffered Chain**

$$t_p = 0.69 \left\lfloor \frac{n}{m} CR_{eq} \frac{m(m+1)}{2} \right\rfloor + \left( \frac{n}{m} - 1 \right) t_{buf}$$

$$= 0.69 \left[ CR_{eq} \frac{n(m+1)}{2} \right] + \left( \frac{n}{m} - 1 \right) t_{buf}$$

$$m_{opt} = 1.7 \sqrt{\frac{t_{pbuf}}{CR_{eq}}}$$

$m_{opt}$ is typically 3 or 4 ..

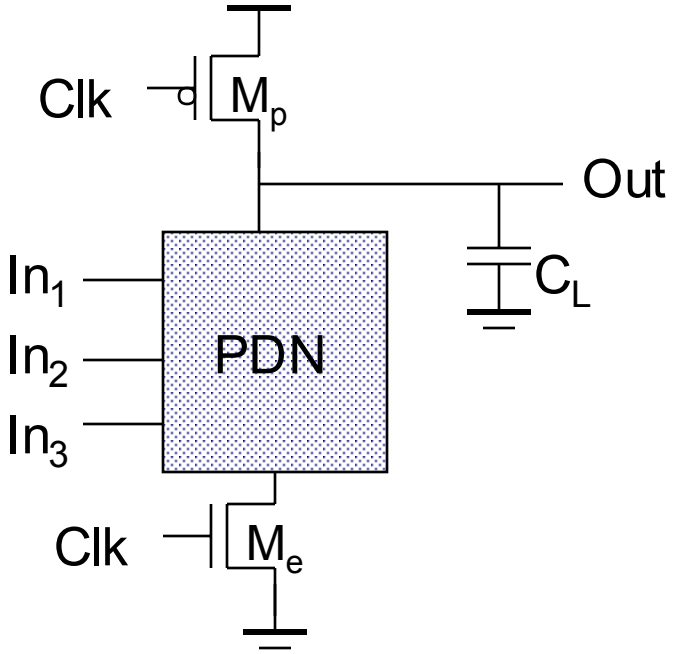
# Dynamic Logic

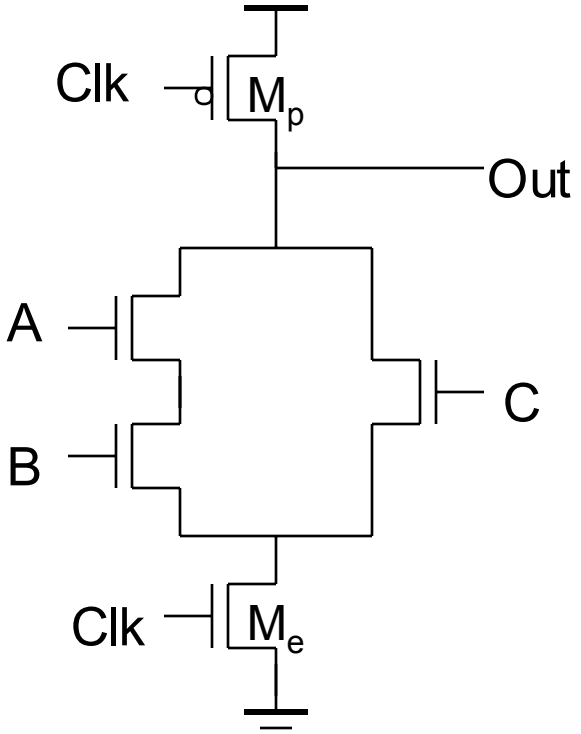
## Dynamic CMOS

- In static circuits at every point in time (except when switching) the output is connected to either GND or $V_{DD}$ via a low resistance path.
  - fan-in of *n* requires 2*n* (*n* N-type + *n* P-type) devices

- Dynamic circuits rely on the temporary storage of signal values on the capacitance of high impedance nodes.
  - requires on *n* + 2 (*n*+1 N-type + 1 P-type) transistors

# Dynamic Gate



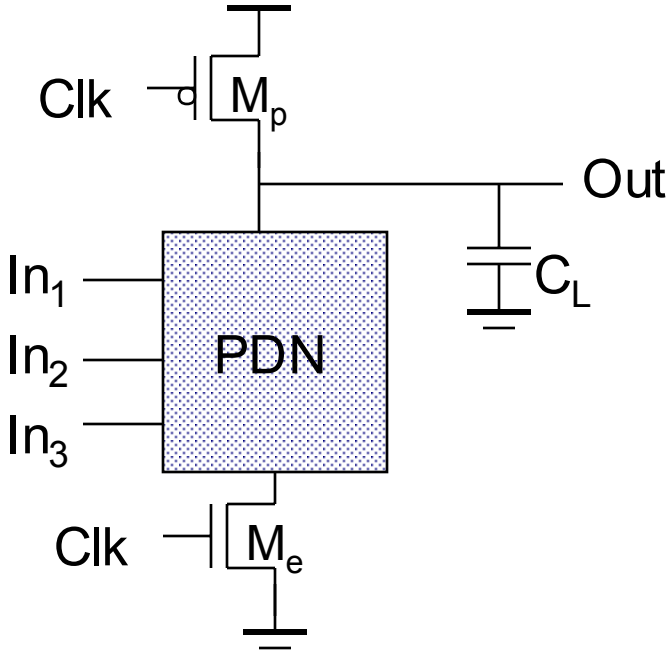



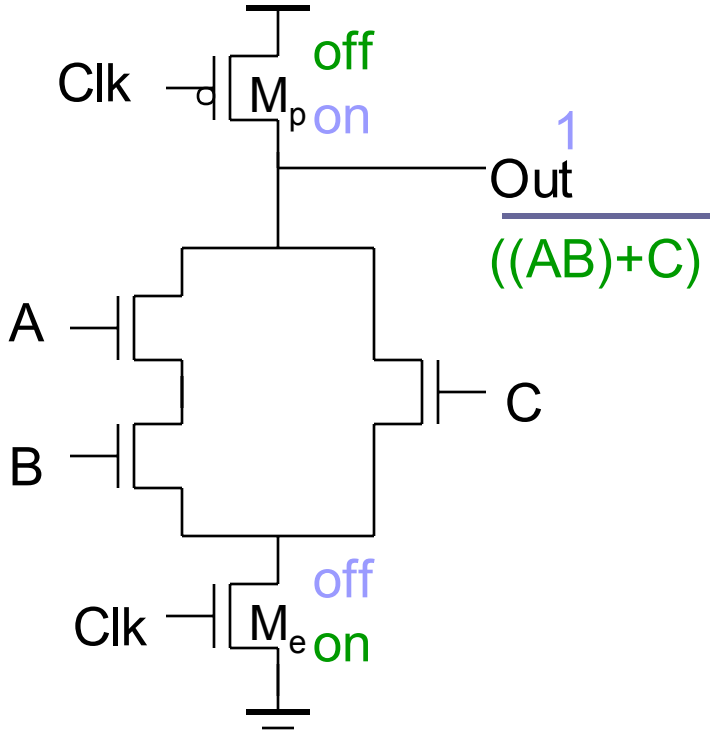

Two phase operation

Precharge (CLK = 0)

Evaluate (CLK = 1)

# Dynamic Gate




Two phase operation

Precharge (Clk = 0)

Evaluate (Clk = 1)

## Conditions on Output

- Once the output of a dynamic gate is discharged, it cannot be charged again until the next precharge operation.
- Inputs to the gate can make at most one transition during evaluation.

- Output can be in the high impedance state during and after evaluation (PDN off), state is stored on $C_L$

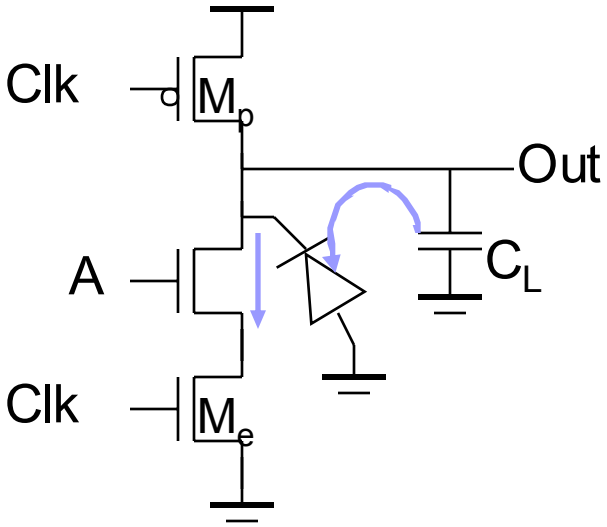
## Properties of Dynamic Gates

- Logic function is implemented by the PDN only
  - number of transistors is N + 2 (versus 2N for static complementary CMOS)
- Full swing outputs (VOL = GND and VOH = VDD)
- Non-ratioed - sizing of the devices does not affect the logic levels
- Faster switching speeds
  - reduced load capacitance due to lower input capacitance (Cin)
  - reduced load capacitance due to smaller output loading (Cout)
  - no Isc, so all the current provided by PDN goes into discharging CL

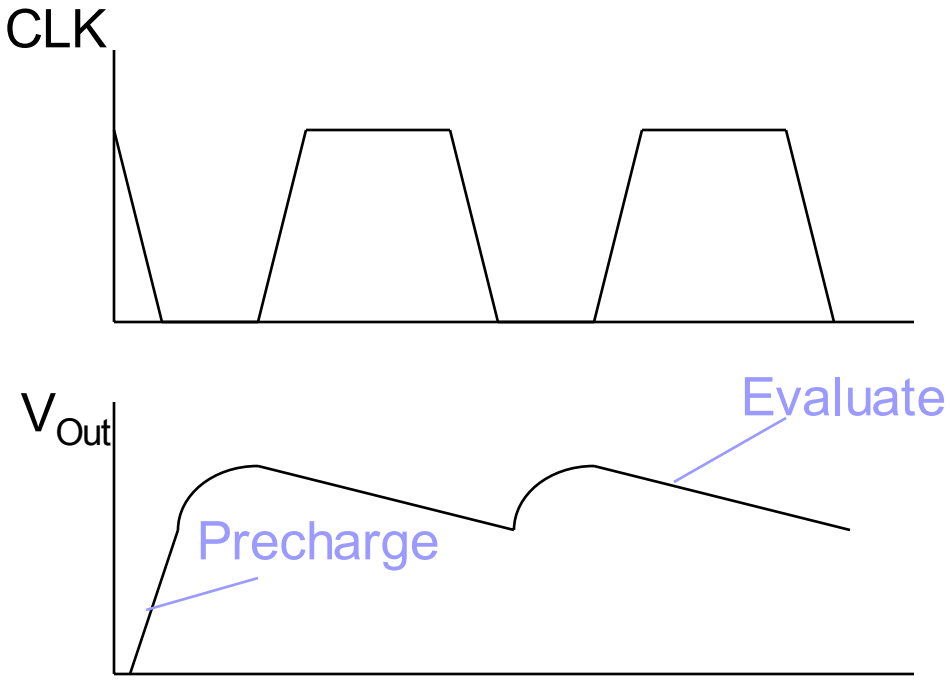
## Properties of Dynamic Gates

- Overall power dissipation usually higher than static CMOS
  - no static current path ever exists between VDD and GND (including Psc)
  - no glitching
  - higher transition probabilities due to periodic charge and discharge
  - extra load on Clk
- PDN starts to work as soon as the input signals exceed VTn, so VM, VIH and VIL equal to VTn
  - low noise margin (NML)
- Needs a precharge/evaluate clock

# Issues in Dynamic Design: Charge Leakage

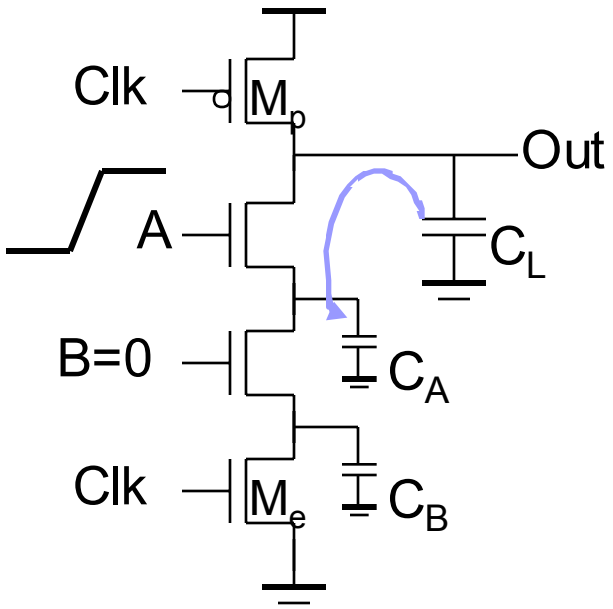



Leakage sources



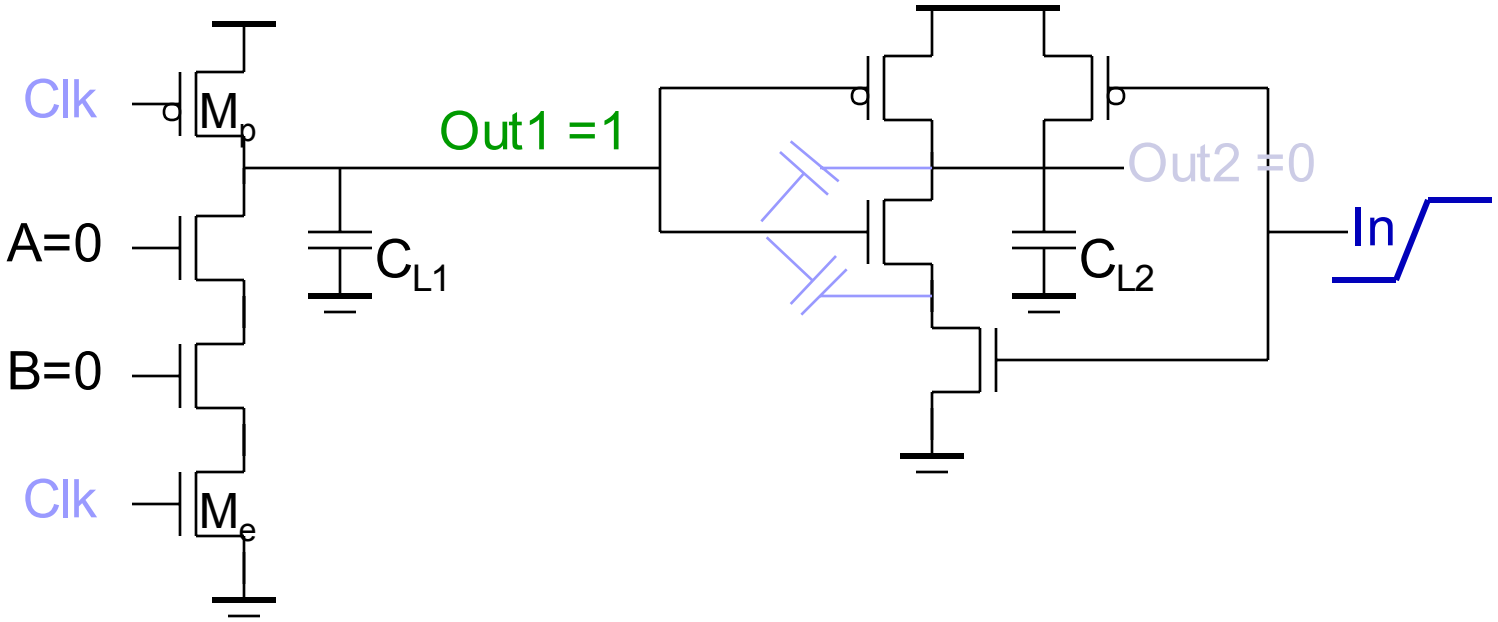

*Dominant component is subthreshold current*

## Issues in Dynamic Design: Charge Sharing




Charge stored originally on $C_L$ is redistributed (shared) over $C_L$ and $C_A$ leading to reduced robustness

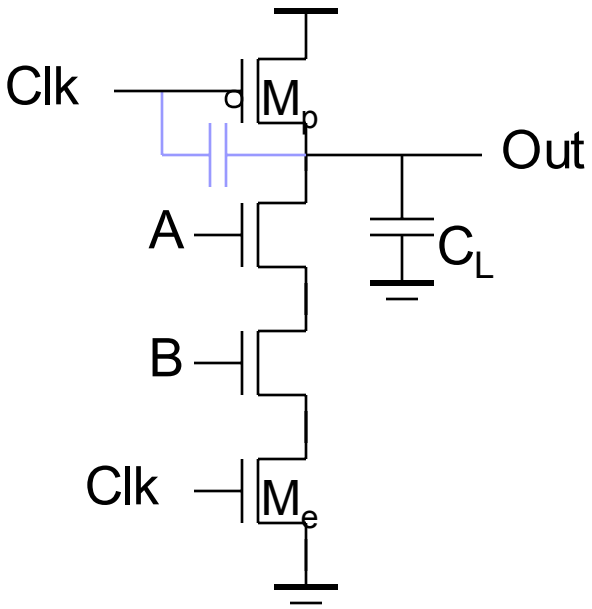
# Issues in Dynamic Design: Backgate Coupling




Dynamic NAND

Static NAND

# Issues in Dynamic Design 4: Clock Feedthrough




Coupling between Out and Clk input of the precharge device due to the gate to drain capacitance. So voltage of Out can rise above $V_{DD}$. The fast rising (and falling edges) of the clock couple to Out.